

Information, Calcul et Communication (SMA/SPH) : Correction de l'Examen final

21 décembre 2023

SUJET 1

INSTRUCTIONS (à lire attentivement)

IMPORTANT! Veuillez suivre les instructions suivantes à la lettre sous peine de voir votre examen annulé dans le cas contraire.

1. Vous disposez de deux heures quarante-cinq minutes pour faire cet examen (8h15 – 11h00).
2. Vous devez **écrire à l'encre noire ou bleu foncée**, pas de crayon ni d'autre couleur.
N'utilisez **pas non plus de stylo effaçable** (perte de l'information à la chaleur).
3. Vous avez droit à toute documentation papier.

En revanche, vous ne pouvez pas utiliser d'ordinateur personnel, ni de téléphone portable, ni aucun autre matériel électronique.

4. Répondez aux questions directement sur la donnée, **MAIS** ne mélangez pas les réponses de différentes questions!

Ne joignez aucune feuilles supplémentaires; **seul ce document sera corrigé.**

5. Lisez attentivement et *complètement* les questions de façon à ne faire que ce qui vous est demandé. Si l'énoncé ne vous paraît pas clair, ou si vous avez un doute, demandez des précisions à l'un des assistants.
6. L'examen comporte 7 exercices indépendants sur 16 pages, qui peuvent être traités dans n'importe quel ordre, mais qui ne rapportent pas la même chose (les points sont indiqués, le total est de 120 points).

Tous les exercices comptent pour la note finale.

Question 1 – Diverses questions [14.5 points]

① [4 points] En supposant que `taille()` est en $\Theta(1)$, quelle est la complexité de l'algorithme ci-contre, où $\lfloor x \rfloor$ représente la partie entière inférieure de x et $L[i : j]$ représente la sous-liste $(L(i), \dots, L(j))$ si $i \leq j$ ou la liste vide si $i > j$?

Justifiez votre réponse.

Réponse et justification :

Cet algorithme est en $\Theta(\log n)$ avec n la taille de la liste en entrée.

En effet, tout est en $\Theta(1)$ sauf les appels récursifs, lesquels séparent toujours la liste en plusieurs parties disjointes dont une seule est traitée à chaque fois. Dans le pire des cas, c'est toujours celle du milieu. Il y a donc $\log_2(n)$ étapes.

On peut aussi faire l'arbre des appels : c'est un arbre ternaire (incomplet et) inéquilibré : $x/4$ pour la première et la troisième branche et $(4-2)x/4$ pour la branche du milieu. De plus seulement **une seule** branche est choisie à chaque fois. Le pire cas étant donc lorsque l'on est toujours au milieu.

On peut aussi écrire simplement l'équation récursive dans le pire des :

$$C(n) = C\left(\frac{4-2}{4}n\right) + \text{cte}$$

algo1
entrée : une liste L non vide
sortie : ??
<pre> n ← taille(L) Si n = 1 Sortir : (L(1))² a ← ⌊ⁿ/₄⌋ Si a = 0 a ← 1 Si L(1) ≥ L(a) b ← algo1(L[1 : a]) Sinon, si L(1) ≥ L(3a) b ← algo1(L[a + 1 : 3a]) Sinon b ← algo1(L[3a + 1 : n]) Sortir : b² </pre>

Commentaire : La complexité est souvent trouvée, mais les justifications ne sont pas toujours correctes ou complètes.

Plusieur(e)s élèves annoncent cependant une complexité linéaire plutôt que logarithmique.

② [3 points] En utilisant RSA, vous souhaitez transmettre de façon confidentielle une information que vous envoyez à un ami dont la clé publique est $(53, 183)$ et dont la clé privée est 77. Votre clé publique est $(97, 201)$ et votre clé privée est 49.

Votre ami reçoit 10001110 (en binaire). Quelle est la valeur déchiffrée par votre ami ?

Exprimez votre réponse sous la forme « $x^y \bmod z$ » (x , y et z en décimal) et justifiez pleinement votre réponse.

Réponse et justification :

Le message encrypté est donc 10001110, soit 142 en décimal.

Vous l'avez encrypté avec la clé *publique* de votre ami, qui lui doit déchiffrer avec sa clé *privée* :

$$142^{77} \bmod 183$$

Commentaire : Excellemment bien réussi ; parfois oublié de justification (pourquoi cette clé ci ?).

③ [5 points] Qu'affiche l'extrait de code ci-dessous ? Si nécessaire, mettez « ??? » pour toute valeur inconnue ou « **Segmentation fault** » si vous pensez que le programme plante (à cet endroit).

Justifiez vos réponses (vous pouvez aussi annoter le code).

```

int i(3);
int* p(&i);
int** q(&p);

**q = 25;
p = &i;

int j(i - 10);

cout << "a. " << *p << endl;
cout << "b. " << j << endl;

i = *p;

int* r(*q);

i = i - *r + 20;
*q = &i;
cout << "c. " << *r - **q << endl;

```

Réponses et justifications :

- a. 25
- b. 15
- c. 0

Pour a. : p pointe sur i à ce stade, dont la valeur est 25 en raison de sa modification au travers de q (qui pointait sur p qui pointait sur i).

Pour b. : j vaut simplement 10 plus la valeur de i à ce moment là, qui est 25.

Pour c. : r pointe vers le même endroit que sur quoi pointait q , c.-à-d. vers le même endroit que p à ce moment là, c.-à-d. vers i , dont la valeur est donc 20 (puisque $i-i+20$).

q pointe toujours vers p qu'on fait toujours pointer vers i , donc $**q$ est la valeur de i .

Le résultat est donc 0.

Commentaire : Pour cette question, qui est pourtant directement tirée d'un quiz sur Coursera, très peu réussissent correctement c). Beaucoup affirment qu'il y a un segmentation fault (pouquoi?). Et les justifications sont très souvent insuffisantes ou inexistantes.

④ [2.5 points] Si un nombre décimal q s'écrit 0.0001 en binaire à virgule fixe, comment s'écrit $\log_2(q)$ en binaire entier signé sur 6 bits? **Justifiez** votre réponse.

Réponse et justification : 111100

q vaut 2^{-4} , donc son log vaut -4 .

Commentaire : trop mettent juste un bit de signe devant l'écriture du positif (pas de complément à 2 donc - est-ce déjà oublié?...)

suite au dos 

Question 2 – Un sac de billes [34 points]

On s'intéresse ici à transmettre les résultats d'un jeu consistant à tirer une bille dans un sac contenant 32 billes de 9 couleurs différentes (toutes présentes au moins une fois).

① [4 points] On commence par considérer le cas particulier où il y a 1 bille rouge, 5 billes de chacune de trois autres couleurs, 2 billes de chacune de deux autres couleurs et 4 billes de chacune des trois dernières couleurs.

Quelle est, en bit, l'entropie du jeu dans ce cas ?

Donnez votre réponse sous la forme $a + b \log_2(5)$, avec a et b deux nombres rationnels ; puis **justifiez** la brièvement.

Réponse et justification :

$$a = \frac{33}{8} \quad \text{et} \quad b = -\frac{15}{32}$$

$$\begin{aligned} & \frac{1}{32} \log_2 \left(\frac{32}{1} \right) + \frac{3 \times 5}{32} \log_2 \left(\frac{32}{5} \right) + \frac{2 \times 2}{32} \log_2 \left(\frac{32}{2} \right) + \frac{3 \times 4}{32} \log_2 \left(\frac{32}{4} \right) \\ &= \log_2(32) - \frac{15}{32} \log_2(5) - \frac{4}{32} \log_2(2) - \frac{12}{32} \log_2(4) \\ &= 5 - \frac{15}{32} \log_2(5) - \frac{1}{8} - \frac{3}{4} = \frac{33}{8} - \frac{15}{32} \log_2(5) \end{aligned}$$

Commentaire : Beaucoup d'erreurs de calculs de tous ordres – vrai aussi pour la suite.

② [3 points] Quelle est l'entropie minimale possible pour un tel jeu ? Expliquez à quelle situation cela correspond, puis **justifiez** brièvement votre réponse.

Réponse et justification :

L'entropie est minimale lorsqu'il y a le moins de choix possibles, c.-à-d. lorsque toutes les couleurs sauf une n'ont qu'une seule bille. La dernière couleur est alors présente $32 - (9 - 1) = 24$ fois.

L'entropie dans ce cas vaut $\log_2(32) - \frac{24}{32} \log_2(24) = 5 - \frac{3}{4} (3 + \log_2(3)) = \frac{11}{4} - \frac{3}{4} \log_2(3)$.

Commentaire : Plusieurs fois oubli de la condition « chaque couleur présente au moins une fois » (ce qui rend la question ⑤ sans aucun intérêt!).

Quelques élèves mélangent entropie maximale et minimale.

③ [4 points] Quelle est l'entropie maximale possible pour un tel jeu ? Expliquez à quelle situation cela correspond, puis **justifiez** brièvement votre réponse.

Réponse et justification :

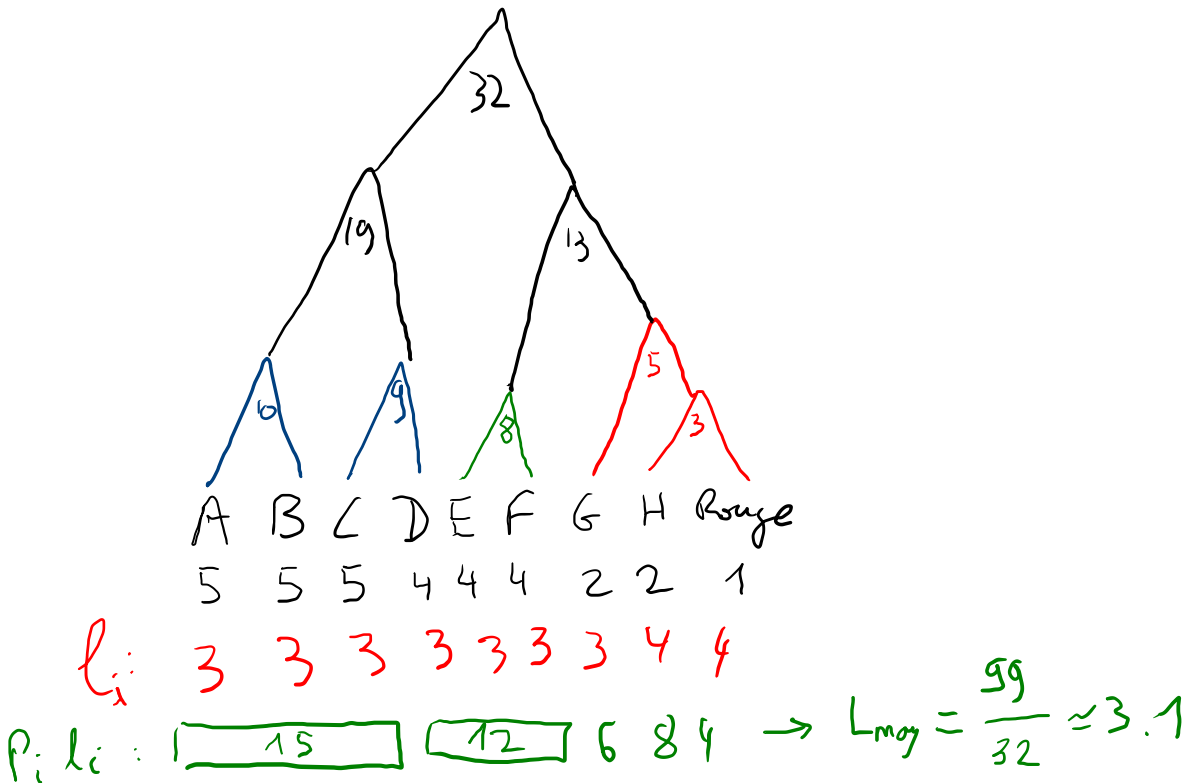
L'entropie est maximale lorsqu'on est le plus équiprobable possible. Or $32/9=3$ et 32 modulo $9=5$. On a donc $9 - 5 = 4$ couleurs à 3 et 5 couleurs à 4.

L'entropie dans ce cas vaut $\log_2(32) - \frac{4 \times 3}{32} \log_2(3) - \frac{4 \times 5}{32} \log_2(4) = 5 - \frac{3}{8} \log_2(3) - \frac{5}{4} = \frac{15}{4} - \frac{3}{8} \log_2(3)$.

④ [5 points] Proposez, en le justifiant, un code de Huffman pour la situation donnée en question ①. Quelle est sa longueur moyenne? **Justifiez** votre résultat.

Réponses et justifications :

Voici un code de Huffman possible :



On commence par le rouge, qui est obligé ; puis le vert ; puis le bleu où **plusieurs choix sont possibles** et on termine par le noir.

On peut aussi avoir la distribution de longueurs suivante :

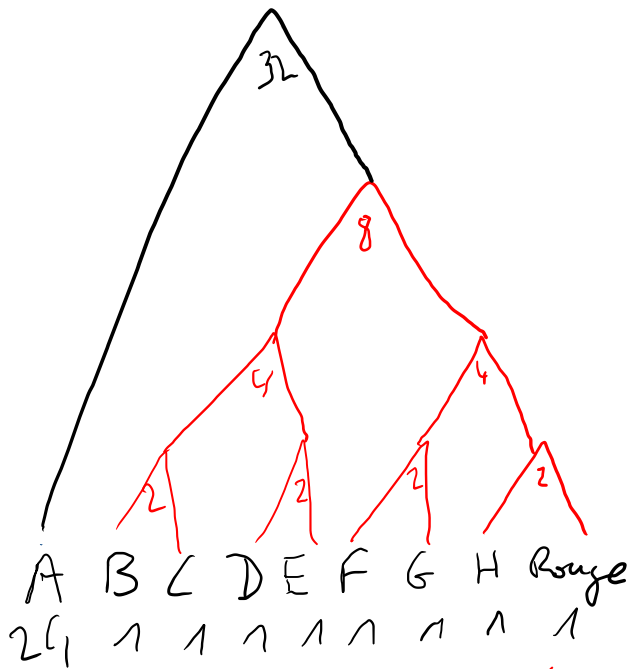
compte	5	5	5	4	4	4	2	2	1
longueur	2	3	3	3	3	3	4	5	5

qui donne (bien sûr) la même longueur moyenne.

Commentaire : Les codes de Huffman sont souvent bien réussis. Plus d'erreurs sur les longueurs moyennes.

⑤ [3 points] Proposez, en le justifiant, un code de Huffman pour la situation donnée en question ②. Quelle est sa longueur moyenne? **Justifiez** votre résultat.

Réponses et justifications :

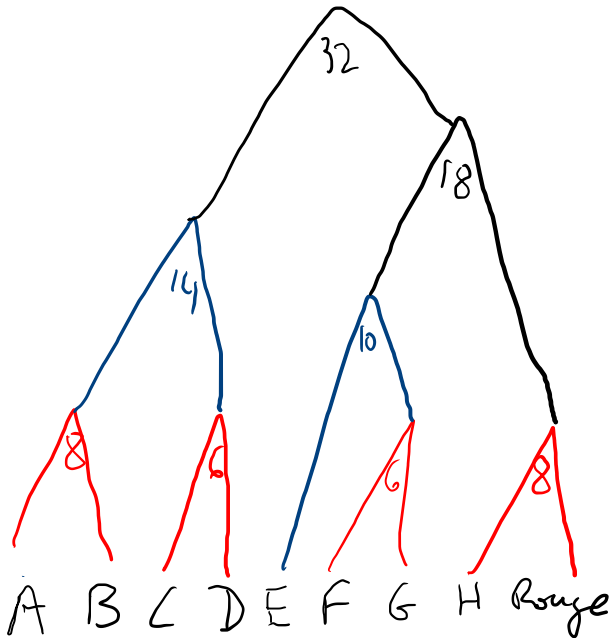


l_i : 1 4 4 4 4 4 4 4 4

$P_i l_i$: 24 4 $\rightarrow L_{\text{moy}} = \frac{56}{32} = \frac{7}{4} = 1.75$

© [5 points] Proposez, en le justifiant, un code de Huffman pour la situation donnée en question ③. Quelle est sa longueur moyenne? **Justifiez** votre résultat.

Réponses et justifications :



l_i : 3 3 3 3 3 4 4 3 3

$P_i l_i$: 12 12 9 9 12 12 12 12 12 $\rightarrow L_{\text{moy}} = \frac{102}{32} \approx 3.19$

⑦ [6 points] Regroupez et discutez pleinement la cohérence de tous les résultats obtenus aux questions ① à ⑥.

Note : quelques valeurs approchées :

$$\log_2(3) \simeq 1.58 > \frac{30}{19}; \quad \log_2(5) \simeq 2.32; \quad \frac{5}{6} \log_2(5) \simeq 1.93; \quad \log_2(7) \simeq 2.81$$

Réponse : On a donc les résultats approchés suivants :

	H	$L(\text{Huffman})$
①	3.0	3.1
②	1.56	1.75
③	3.15	3.2

pour ① : $\frac{33}{8} - \frac{15}{32} \log_2(5) \simeq 4 + \frac{1}{8} - \frac{15 \times 2.32}{32} \simeq 4 + \frac{1}{8} - \frac{32}{32} - \frac{5}{32} \simeq 3$

pour ② : $\frac{11}{4} - \frac{3}{4} \log_2(3) \simeq \frac{11-4.74}{4} \simeq \frac{6.26}{4} \simeq 1.56$

pour ③ : $\frac{15}{4} - \frac{3}{8} \log_2(3) \simeq \frac{80-4.74}{8} \simeq \frac{25.26}{8} \simeq 3.15$ ou 3.16

On peut vérifier dans chaque cas que :

— l'entropie est inférieure à $\log_2(9) \simeq 2 \times 1.58 = 3.16$

Pour le cas ③, on peut vérifier plus précisément : $2 \log_2(3) - \frac{15}{4} + \frac{3}{8} \log_2(3) = \frac{19}{8} \log_2(3) - \frac{30}{8} = \frac{19}{8} (\log_2(3) - \frac{30}{19}) > 0$

— $H \leq L(\text{Huffman})$

— $L(\text{Huffman}) < H + 1$

De plus, on a bien $H(②) \leq H(①) \leq H(③)$.

Tous nos résultats semblent donc cohérents.

Commentaire : Trop peu d'élèves (mais il y en a !) pensent à tout vérifier. Sachez utiliser *toutes* vos connaissances.

⑧ [4 points] On cherche finalement à transmettre les résultats de 7 tirages. Quel taux de compression peut-on espérer en moyenne par rapport à un codage binaire direct de chacune des 9 couleurs ?

Réponse : Si l'on codait directement en binaire chaque couleur, il faudrait 4 bits par couleur.

D'après les résultats précédents, on peut espérer transmettre les couleurs (en utilisant les codes de Huffman) entre 1.75 et 3.2 bits par couleur, soit un gain/taux de compression entre 56% ($\frac{4-1.75}{4}$) et 20% ($\frac{4-3.2}{4}$).

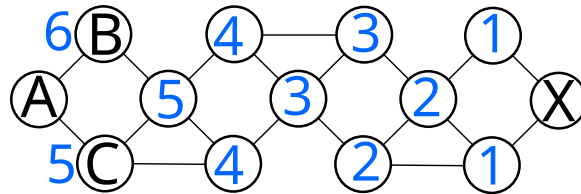
Note : le nombre de tirages à transmettre est totalement inutile ici.

Commentaire : Trois gros types d'erreurs ici :

1. prendre l'entropie comme critère alors que justement les longueurs optimales (codes de Huffman) ont été calculées ;
2. faire divers mélanges entre les différentes situations (dont celle de ① qui n'a rien de générale) alors que les situations extrêmes sont connues ;
3. ne pas calculer le taux de compression, mais le « taux de préservation » (ce qui reste, au lieu de calculer ce qui est enlevé)

Question 3 – La bonne route [8.5 points]

On considère le réseau (informatique) suivant :



① [4 points] Donnez la ligne correspondant à X dans la table de routage de A.

Justifiez votre réponse en annotant (simplement) le graphe ci-dessus.

Réponse :

Destination	Direction	Distance
X	C	6

Commentaire : Trop d'élèves ne justifient pas ou pas correctement : pourquoi cette route là plutôt qu'une autre (comment a-t-elle été trouvée) ?

② [1.5 points] À quelle couche (*nom*) correspond le protocole TCP ? Le protocole IP ?

Réponses : TCP : transport (couche 4) ; IP : réseau (couche 3).

③ [3 points] Si ce réseau perd des paquets, que se passe-t-il :

- au niveau du protocole IP pour la communication entre A et B ?
- au niveau du protocole TCP pour la communication entre A et X ?

Réponses : rien au niveau IP (ce n'est pas géré au niveau de cette couche).

Au niveau TCP, A va attendre les paquets d'aquittement de X et retransmettre si nécessaire.

Commentaire : Plusieurs n'ont pas compris ce qui se passe vraiment au niveau IP.

suite au dos ➡

Question 4 – Duplicata [12 points]

Ecrivez un programme C++ qui permet de recopier toutes les lignes d'un fichier dans un autre. Le programme demandera le nom du fichier à lire, puis le nom du fichier à écrire et recopiera chacune des lignes du premier fichier dans le second.

Réponse :

Voici une version possible :

```
#include <iostream>
#include <fstream>
#include <string>
using namespace std;

string demander_nom(const string& verbe)
{
    cout << "Entrez un nom de fichier à " << verbe << " : ";
    string nom;
    cin >> nom; // ou getline(cin, nom);
    return nom;
}

void erreur(const string& nom, const string& mode)
{
    cerr << "Impossible d'ouvrir le fichier \""
         << nom << "\" en " << mode << endl;
}

void copie(const string& nom_lecture, const string& nom_ecriture)
{
    ifstream entree(nom_lecture);
    ofstream sortie(nom_ecriture);
    if (entree.fail()) erreur(nom_lecture, "lecture");
    else if (sortie.fail()) erreur(nom_ecriture, "écriture");
    else {
        string ligne;
        while (getline(entree, ligne)) sortie << ligne << endl;
        entree.close(); // optionnels
        sortie.close();
    }
}

int main()
{
    copie(demander_nom("lire"), demander_nom("écrire"));
    return 0;
}
```

Voici une autre version, similaire :

```
#include <iostream>
#include <fstream>
#include <string>
using namespace std;

string demander_nom(const string& verbe)
{
    cout << "Entrez un nom de fichier à " << verbe << " : ";
    string nom;
    cin >> nom; // ou getline(cin, nom);
    return nom;
}

void erreur(const string& nom, const string& mode)
{
    cerr << "Impossible d'ouvrir le fichier \""
         << nom << "\" en " << mode << endl;
}

int main()
{
    string temp(demander_nom("lire"));
    ifstream entree(temp);
    if (entree.fail()) {
        erreur(temp, "lecture");
        return 1;
    }

    temp = demander_nom("écrire");
    ofstream sortie(temp);
    if (sortie.fail()) {
        erreur(temp, "écriture");
        return 1;
    }

    while (getline(entree, temp)) {
        sortie << temp << endl;
    }

    entree.close(); // optionnels
    sortie.close();

    return 0;
}
```

Et voici une *mauvaise* façon de faire juste (n'a pas tous les points) :

```
#include <iostream>
#include <fstream>
#include <string>
using namespace std;

int main()
{
    cout << "Entrez un nom de fichier à lire : ";
    string nom;
    cin >> nom; // ou getline(cin, nom);
    ifstream entree(nom);
    if (entree.fail()) {
        cout << "Impossible d'ouvrir le fichier \""
            << nom << "\" en lecture" << endl;
        return 0;
    }

    cout << "Entrez un nom de fichier à écrire : ";
    cin >> nom; // ou getline(cin, nom);
    ofstream sortie(nom);
    if (sortie.fail()) {
        cout << "Impossible d'ouvrir le fichier \""
            << nom << "\" en écriture" << endl;
        return 0;
    }

    while (getline(entree, nom)) {
        sortie << nom << endl;
    }

    return 0;
}
```

Commentaire : Plusieurs ne donnent pas un programme complet (comme demandé).

Pas encore assez de modularisation (= trop de « copiés-collés »)

Plus mineur : attention (en C++) `getline()` ne lit pas le retour à la ligne (contrairement à la version C, pour celles/ceux qui connaîtraient ce langage!).

suite au dos 

Question 5 – Quelques algorithmes [12 points]

① [6 points] Écrivez un algorithme *récuratif* et *sans* boucle permettant de supprimer d'une liste donnée toutes les valeurs strictement plus petites qu'une valeur donnée (sans changer l'ordre).

Par exemple, pour la liste (2, 12, 7, 9, 3, 5, 18) et la valeur 9, l'algorithme devra sortir (12, 9, 18); et pour cette même liste et la valeur 19, l'algorithme devra sortir la liste vide.

Réponse :

Il y a plein de façons d'écrire un tel algorithme, par exemple :

algo1
entrée : liste L et valeur x sortie : liste L' [...]
Si L est vide Sortir : () // liste vide
$n \leftarrow$ taille (L) // pourrait être passé comme paramètre
Si $L(1) < x$ Sortir : algo1 (($L(2), \dots, L(n)$), x) // sous-liste
Sortir : $L(1) \oplus$ algo1 (($L(2), \dots, L(n)$), x)

avec \oplus l'ajout d'un élément en début de liste.

On peut aussi couper la liste en 2 au milieu, ou en 3 comme dans la question 1, ou peu importe tant que c'est complet et gardé dans l'ordre.

Commentaire : Encore beaucoup de difficultés à écrire des algorithmes propres; en particulier la gestion de la récursion :

- les cas terminaux sont souvent oubliés / mal gérés; typiquement le cas de la liste vide n'est pas toujours traité;
- utilisation assez fréquente de listes externes/globales (qui seraient les mêmes entre les appels récursifs) non initialisées/déclarées pour stocker des valeurs/le résultat;
- utilisation de listes locales non initialisées/déclarées;
- oubli du « **Sortir** » lors de l'appel récursif;
- oubli d'utiliser (typiquement concaténer) le résultat de l'appel récursif;
- utilisation de la notation \oplus sans la définir (alors que l'on a insisté plusieurs fois en séances sur l'*ambiguïté* d'une telle notation).


② [6 points] Écrivez un algorithme *récuratif* et *sans* boucle qui, à partir de deux listes de même taille, crée la liste (de même taille) contenant à chaque position la plus grande des deux valeurs des deux listes à cette même position.

Par exemple, pour les listes (2, 7, 12, 9, 18, 5) et (5, 2, 17, 9, 3, 4), l'algorithme devra sortir (5, 7, 17, 9, 18, 5).

Réponse :

Il y a là aussi plein de façons d'écrire un tel algorithme, par exemple :

algo2
entrée : deux listes L_1 et L_2 de même taille sortie : liste L' [...]
Si L_1 est vide Sortir : () // liste vide $a \leftarrow L_2(1)$ Si $L_1(1) > L_2(1)$ $a \leftarrow L_1(1)$ $n \leftarrow \text{taille}(L_1)$ // pourrait être passé comme paramètre Sortir : $a \oplus \text{algo1}((L_1(2), \dots, L_1(n)), (L_2(2), \dots, L_2(n)))$

suite au dos 

Question 6 – Bon voyage ! [29 points]

Note : (comme toujours, mais particulièrement ici) nous vous conseillons, afin de faire les bons choix, de lire *entièrement* cette question avant de commencer.

On s'intéresse ici à écrire des *parties* d'un programme C++ permettant de représenter un réseau de transport. Le réseau que nous souhaitons représenter est un ensemble de gares, dont certaines sont reliées par des liaisons ayant une durée (en minutes) et un prix (en francs).

Chaque gare aura un nom et l'ensemble de ses destinations (c.-à-d. des liaisons : durée, prix, et indication de la gare d'arrivée).

① [4 points] Définissez les types de données que vous jugez nécessaires pour représenter le problème spécifié ci-dessus.

Réponse :

```
typedef unsigned int Minutes; // optionnel

struct Liaison {
    Minutes temps;
    double prix;
    size_t arrivee;
};

struct Gare {
    string nom;
    vector<Liaison> destinations;
};

typedef vector<Gare> Reseau;
```

L'arrivée de la Liaison peut aussi être un `const Gare&`, ou un pointeur, mais c'est alors plus compliqué pour la fonction `connexions()` de la fin, car il faut retrouver l'indice de la gare en question. Ce ne peut *en aucun cas* être directement une `Gare`.

Commentaire : Plusieurs oublis de définir le réseau.

Quelques tentatives d'utiliser les pointeurs (ce qui serait vraiment bien ici), mais souvent mal gérés/-manipulés. Ce point (pointeurs) devrait vraiment être revu avant le prochain semestre.

② [4 points] Écrivez le code C++ permettant de représenter le réseau simple constitué de deux gares, Lausanne et Renens, tel que la liaison Lausanne–Renens prend 6 minutes et coûte 2.45 francs, et la liaison Renens–Lausanne prend 7 minutes et coûte 2.20 francs.

Réponse :

```
Reseau cff({
    { "Lausanne", {{ 6, 2.45, 1 }} },
    { "Renens" , {{ 7, 2.20, 0 }} },
});
```

③ [5 points] Écrivez une fonction `prix()` qui permet de tester si un ensemble de liaisons contient une gare donnée. Cette fonction retournera un booléen indiquant si l'ensemble de liaisons contient la gare fournie et, si oui, le prix de cette liaison (sinon un prix quelconque).

Vous êtes libre de choisir les arguments qui vous conviennent le mieux pour cette fonction.

Réponse : Par exemple :

```

struct Prix_valide {
    bool valide;
    double prix;
};

constexpr Prix_valide invalide({ false, 0.0 });

Prix_valide prix(vector<Liaison> const& liste, size_t indice)
{
    // recherche linéaire
    // ils peuvent utiliser find() s'ils connaissent
    for (auto const& element : liste) {
        if (element.arrivee == indice) return { true, element.prix };
    }
    return invalide;
}

```

Bien sûr cela peut changer en fonction de ce qui est contenu dans une `Liaison` pour représenter une gare. On peut supposer si nécessaire que le nom d'une gare est non ambigu (c.-à-d. est un identifiant unique de cette gare).

Pour le type de retour, on peut aussi simplement retourner un `bool` et passer le prix par référence.

④ [1 point] Définissez un type `Trajet` qui permet de représenter un trajet dans le réseau, c.-à-d. un ensemble de gares à parcourir. Vous êtes libre de choisir la représentation qui vous convient le mieux.

Réponse :

```

typedef vector<size_t> Trajet;

```

Ce ne doit pas être un tableau de `Gare`, mais bien un tableau de moyen de référencer des `Gare` (indice, ou pointeur, ou nom).

Ce peut aussi être une `struct` qui contient un `vector`.

Commentaire : Une très grande majorité n'a pas tiré les leçons du homework et a choisi un tableau de gares. La notion d'indirection et d'éviter les copies n'est donc pas assimilée en soi (n'est pas un réflexe).

⑤ [10 points] Écrivez une fonction `prix()` qui permet de calculer le prix total d'un trajet donné. Cette fonction retournera un booléen indiquant si le trajet est valide ou non et son prix total (si c'est le cas, sinon une valeur quelconque).

Pour être considéré comme valide, un trajet doit contenir un ensemble de gares tel que chaque gare est contenue dans l'ensemble des destinations de la gare qui la précède dans le trajet.

Vous êtes libre de choisir les arguments qui vous conviennent le mieux pour cette fonction.

Réponse :

```

Prix_valide prix(Trajet const& t, Reseau const& r)
{
    if (t.empty()) return invalide;

    size_t last(t[0]);
    if (t[0] >= r.size()) return invalide; // bonus

    Prix_valide retour({ true, 0.0 });
    for (size_t i(1); i < t.size(); ++i) {
        const auto [ ok, p ] = prix(r[last].destinations, t[i]);
        if (not ok) return invalide;
        retour.prix += p;
        last = t[i];
    }
    return retour;
}

```

A noter que si dans `Liaison arrivee` est un pointeur, il n'est pas nécessaire de passer le `Reseau` en argument ici.

Commentaire : Trop peu pensent à vérifier que le trajet n'est pas vide.

© [5 points] Pour pouvoir appliquer des algorithmes de plus court chemin (sur les durées), il est nécessaire de produire une matrice représentant les durées entre chacune des gares du réseau :

- à la position (i,j) de la matrice sera stockée la durée de la liaison de la *i*-ème gare à la *j*-ème gare;
- si les deux gares ne sont pas connectées, on mettra `numeric_limits<X>::max()`, où *X* est le type que vous avez choisi pour les durées;
- et on mettra 0 sur la diagonale.

Par exemple, si Lausanne est la première gare et Renens la seconde dans l'exemple donné en sous-question ②, la matrice à produire serait

$$\begin{pmatrix} 0 & 6 \\ 7 & 0 \end{pmatrix}$$

Écrivez une fonction `connexions()` qui prend en paramètre un réseau et qui retourne la matrice correspondante, telle que définie ci-dessus.

Réponse :


```

typedef vector<vector<Minutes>> Matrice;

Matrice connexions(Reseau const& r)
{
    const size_t n(r.size());
    Matrice retour(n, vector<Minutes>(n, numeric_limits<Minutes>::max()))
    for (size_t i(0); i < n; ++i) {
        for (auto l : r[i].destinations) {

            const size_t j = l.arrivee;
            /* ou, suivant la conception :
             * j = recherche(l, r);          */

            retour[i][j] = l.temps;
        }
        retour[i][i] = 0.0;
    }
    return retour;
}

```

Les typedef sont une bonne pratique, optionnelle ici (mais pensez-y!).

Question 7 – Enregistrement [10 points]

On considère le signal suivant :

$$X(t) = \sum_{i=1}^5 \frac{1}{i^2} \sin(2\pi i f t + (i-1)\frac{\pi}{10})$$

avec $f = 210$ Hz.

Ce signal est filtré par un filtre passe-bas idéal de fréquence de coupure $f_c = 800$ Hz. On désire enregistrer sous forme numérique une seconde de ce signal filtré. Pour cela, on l'échantillonne ; chaque échantillon étant représenté sur 12 bits.

- ① [1 point] Quelle est la bande passante de X (avant filtrage) ? **Réponse :** $f_{\max} = 1050$ Hz.
- ② [6 points] Combien de bits au total faut-il enregistrer *au minimum* si l'on veut éviter d'avoir des erreurs lors de la reconstruction du signal enregistré ?

Justifiez pleinement votre réponse.

Note : les fréquences d'échantillonnage peuvent être réglées au hertz près.

Réponse et justification :

Le filtrage à 800 Hz enlève donc les composantes $i = 4$ (840 Hz) et $i = 5$ (1050 Hz). La bande passante du signal échantillonné est donc 630 Hz ($i = 3$).

Il faut donc échantillonner au minimum à 1261 Hz (*strictement* au dessus de deux fois la bande passante). Ce qui, pour une seconde, donne $12 \times 1261 \simeq 15$ kbits.

- ③ [3 points] Quelle est alors la forme (formule mathématique) du signal reconstruit ?

Justifiez *brièvement* votre réponse.

Réponse :

Puisque le signal ainsi échantillonné est parfaitement reconstruit (c'est ce qu'on veut), il correspond au signal avant filtrage, soit :

$$\sum_{i=1}^3 \frac{1}{i^2} \sin(2\pi i f t + (i-1)\frac{\pi}{10})$$

avec $f = 210$ Hz.

Commentaire : Cet exercice est en général très bien réussi dans la globalité.

L'erreur la plus fréquente étant de donner l'inégalité stricte, mais de prendre au final une fréquence d'échantillonnage *égale* à deux fois la bande passante.