

---

# DM2

---

03 janvier 2025

## 1 Questions

### Questions 1

Pour  $i$ , l'indice de l'échantillon, sa valeur est  $U(i\tau_{ech})$ . Or, pour  $U(t) = U_0 \sin 2\pi ft$ ,  $U(i\tau_{ech}) = U_0 \sin(2\pi f i\tau_{ech})$ .

### Question 3

```
1  52 49 46 46 2e 00 00 00 57 41 56 45 66 6d 74 20
2  10 00 00 00 01 00 01 00 22 56 00 00 44 ac 00 00
3  02 00 10 00 64 61 74 61 0a 00 00 00 d2 03 5e 06
4  ff ff a2 f9 ff 7f
```

### Question 18

On note  $n$  le nombre de pistes et  $l_1, \dots, l_n$  le nombre d'échantillons total de chaque piste. Tout d'abord, déterminons la complexité de `reduce_track`. Toutes les opérations hormis les boucles sont en  $O(1)$ .

Première boucle :

```
1  for (int i = 0; i < t->n_sounds; i++){
2      s->n_samples += t->sounds[i]->n_samples;
3  }
```

Deuxième boucle :

```
1  for (int i = 0; i < t->n_sounds; i++){
2      for (int j = 0; j < t->sounds[i]->n_samples; j++){
3          samples[n] = t->sounds[i]->samples[j];
4          n++;
5      }
6  }
```

On observe que la première boucle est un grand  $O$  de la deuxième puisque cette dernière possède une sous-boucle que la première n'a pas. Or, la deuxième boucle parcourt tous les échantillons de la piste. Elle est donc en  $O(l_k)$ . La complexité de `reduce_track` est donc aussi  $O(l_k)$ .

Pour ce qui est de `reduce_mix` :

Première boucle :

```

1   for (int i = 0; i < m->n_track; i++){
2       n = 0;
3       for (int j = 0; j < m->tracks[i]->n_sounds; j++) {
4           n += m->tracks[i]->sounds[j]->n_samples;
5       }
6       if (n > max_samples){
7           max_samples = n;
8       }
9   }

```

Elle parcourt tous les sons de chaque piste.

```

1   int* samples_sum = calloc(max_samples, sizeof(int));

```

Est un grand  $O$  de **max\_samples** qui est évidemment un grand  $O$  de la première boucle.

Deuxième boucle :

```

1   for (int i = 0; i < m->n_track; i++){
2       s_temp = reduce_track(m->tracks[i]);
3       for (int j = 0; j < s_temp->n_samples; j++){
4           samples_sum[j] += s_temp->samples[j] * m->vols[i];
5       }
6       free_sound(s_temp);
7   }

```

On rappelle que **reduce\_track** est un grand  $O$  de  $l_k$ . Et, la sous-boucle parcourt tous les échantillons de la piste. Elle est donc aussi un grand  $O$  de  $l_k$ . La boucle extérieure parcourt toutes les  $n$  pistes. Donc l'ensemble est de complexité :  $O(l_1 + \dots + l_n)$ . On note que la première boucle de la fonction parcourt moins d'éléments que la deuxième. Donc, est aussi un grand  $O$  de  $l_1 + \dots + l_n$ . Pour finir, la dernière boucle est un grand  $O$  de **max\_samples**, donc aussi de  $l_1 + \dots + l_n$ . Au final, **reduce\_mix** est de complexité :  $O(l_1 + \dots + l_n)$ .