DM2

Élie Bellot des Minières

06 octobre 2024

1 Questions

Questions 1

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

Question 3

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

Question 18

On note n le nombre de pistes et $l_1, \ldots, l_n$ 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 :

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

Deuxième boucle :

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

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 :

```
for (int i = 0; i < m->n_track; i++){
    n = 0;
    for (int j = 0; j < m->tracks[i]->n_sounds; j++) {
        n += m->tracks[i]->sounds[j]->n_samples;
    }
    if (n > max_samples){
        max_samples = n;
    }
}
```

Elle parcourt tous les sons de chaque piste.

```
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 :

```
for (int i = 0; i < m->n_track; i++){
    s_temp = reduce_track(m->tracks[i]);
    for (int j = 0; j < s_temp->n_samples; j++){
        samples_sum[j] += s_temp->samples[j] * m->vols[i];
    }
    free_sound(s_temp);
}
```

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 : $O(l_1 + \ldots + l_n)$. On note que la première boucle de la fonction parcourt moins d'éléments que la deuxième. Donc, est aussi un grand O de $l_1 + \ldots + l_n$. Pour finir, la dernière boucle est un grand O de `max_samples`, donc aussi de $l_1 + \ldots + l_n$. Au final, `reduce_mix` est de complexité : $O(l_1 + \ldots + l_n)$.