Benzekri Omar
300262795

24/09/2021

uOttawa
L'Université canadienne
Canada's university

Devoir 2 (2.5% - 10 points)

CSI2110/CSI2510 (Automne 2022)

Question 1. **(4 points)**

a) (2 points) Quelle sera la sortie produite par cet algorithme pour le tableau montré?

Pour le tableau A montré, la sortie sera de 29 car le premier loop va d'abord ajouter uniquement les nombres impairs au stack S et puis le deuxième loop va les additionner en t qui aura une valeur de 29 quand le temps viendra de l'imprimer.

b) (2 points) Quelle est la complexité en termes de Grand O de cet algorithme en fonction de n? Une brève justification est suffisante.

La complexité en termes de Grand O de cet algorithme en fonction de n sera O(n) car l'algorithme doit parcourir le tableau complet pour découvrir tous les nombres impairs.

Question 2. **(6 points)**

We're assuming that all the kinds are Queues that are in a ArrayList called "P" and that there is another queue of all the pizzas. The integer kind will be used as an index.

```
addPizza(kind)
{
        Pizza s = new Pizza();
        s.setTime(system.currentTime);
        (P.get(kind)).add(s)
}
```

This operation has a running time of O(1), since it only needs to create and add the new pizza to the kind's queue.

```
getPizza(kind)
{
        return ((P.get(kind)).remove() )
}
```

This operation has a running time of O(1) since ir only needs to get the head of kind's queue.

```
getSurprisePizza()
{
        int oldest = 0;
        for (int i=0; i<(P.size()-1);i++)
        {
                if (((P.get(i)).peek()).getTime()>((P.get(oldest)).peek()).getTime())
                {
                oldest = i;
                }
        }
        return ((P.get(i)).remove())
}
```

This operation has a running time of O(M), M being the number of kinds there are. This method needs to compare the head of all the kinds to find the oldest and then return it.