

## Homework 11

ALECK ZHAO

December 1, 2017

### 1 Bin Packing (50 points)

Suppose that we are given a set of  $n$  objects, where the size  $s_i$  of the  $i$ th object satisfies  $0 < s_i < 1$ . We wish to pack all the objects into the minimum number of unit-size bins. Each bin can hold any subset of the items whose total size does not exceed 1. In other words, we are trying to find a function  $f : [n] \rightarrow [k]$  so that  $\sum_{i:f(i)=j} s_i \leq 1$  for all  $j \in [k]$ , and our goal is to minimize  $k$ . Let  $S = \sum_{i=1}^n s_i$ .

- (a) Prove that the optimal number of bins required is at least  $\lceil S \rceil$ .

The *first-fit algorithm* considers each object in turn (from 1 to  $n$ ) and places it in the first bin that can accommodate it. If there is no such bin, then we create a new bin for it and make it the last bin. Note that this defines an ordering over bins based on when we created them, so “first” and “last” make sense.

- (b) Prove that the first-fit algorithm leaves at most one bin at most half full. In other words, all bins but one are more than half full.
- (c) Prove that the first-fit algorithm is a 2-approximation to bin packing.

### 2 Pirate Treasure (50 points)

You are currently standing on an east-west road next to a sign. This sign tells you that somewhere along the road there is a pirate treasure that is sitting in plain sight and yet which is currently unclaimed. Unfortunately, the sign does not say whether the treasure is to the east of the sign or to the west, and also does not tell you the distance from the treasure. Your goal is to find the treasure while minimizing the amount that you travel. Let  $x$  denote the distance from the sign to the treasure (which you do not know). So if you walk in the correct direction and never turn around, your cost will be  $x$  (since you will find the treasure after walking distance  $x$ ). However, if you walk in the *incorrect* direction and never turn around, then your cost will be infinite (since you will keep walking forever). You may assume that  $x$  is an integer. T

- (a) Design a deterministic  $O(1)$ -competitive algorithm for this problem. That is, design an algorithm which finds the treasure and involves walking a total distance of at most  $O(1) \cdot x = O(x)$ . Prove the competitive ratio of your algorithm.

Now suppose that instead of the sign being located somewhere along an east-west road, it is located at an intersection where you have  $m$  directions to choose from (so  $m$  was 2 in the previous part). The sign tells you that the treasure is somewhere along one of the roads, but doesn't tell you which road or the distance to the treasure.

- (b) Design a deterministic  $O(m)$ -competitive algorithm for this problem (and prove that it is  $O(m)$ -competitive).
- (c) Prove that for any deterministic algorithm, there is an input (location of the treasure) so that the competitive ratio of the algorithm is  $\Omega(m)$ .