Elissa Ito December 10, 2022

6.1210 Problem Set 7

Problem 1 *(Collaborators: None)*

a) Each container takes exactly 1 unit of time to eat. The optimal solution will have no idle time, since if there were, the interval could be "pushed" to begin at the finishing time of the previous interval. Thus the first eating interval will begin at time 0 and end at 1, the next from 1 to 2, and so on, thus all leftovers are eaten at integral times.

b) By a), a container gets eaten for every integer of time from 0 to the completion time. So if the container with highest happiness has index j with expiration time t_j and happiness h_j, there is an integer m where $m+1 \leq t_j \leq m+2$ (i.e. the container with the happiness gets eaten from time m to $m+1$). Out of all the containers that have an expiration time $m+1 \leq t_i \leq m+2$, eating the container with the highest happiness to be eaten during the interval from m to $m+1$ is optimal for maximum happiness.

c) 1) Order containers from smallest t_i to largest t_i. Let this list be called C, $C = c_1, c_2, ...c_n$ where the expiration of c_i is smaller than or equal to that of c_{i+1} (i.e. $t_i \leq t_{i+1}$).

2) Eat the container with the smallest t_i during the time interval from 0 to 1. Current time, t_c is now 1. Current list is $C' = C \setminus \{c_1\}$

3) Find first container in S has t_i such that $t_i > t_c + 1$, eat the next container. Current time is $t_c = t_c + 1$. Current list is $C' = C' \setminus \{c_i\}$. If not, eat the next container, making the current time $t_c = t_c + 1$.

4) Repeat step 3 until no containers are left. The containers you ate is optimal by the greedy algorithm.

Greedy choice property: if one has the choice to eat or not eat a container, it is optimal to eat is as h_i is nonnegative.
Runtime: Sorting the list takes $O(nlogn)$ time, and iterating through the list takes $O(n)$ time. The algorithm takes $O(n)$ time.
Let S be an optimal set of containers and G be the greedy set. Assume $|S| > |G|$. By the Greedy Choice Property, we can assume that $g_1 \epsilon S$. Let $A' \subset A$ be the set of containers that start after g_1 ends. By Strong Induction, $G \setminus \{g_1\}$ is optimal for A'. Hence $|G| - 1 \leq |S| - 1$ and so $|G| \geq |S|$, a contradiction.

Problem 2 *(Collaborators: None)*

1. Sort intervals by earliest starting time to latest starting time.
2. Begin by designating the person with the earliest starting interval as the safety contact.
3. Among all the following intervals with later starting times and earlier finish time than the current safety contact's interval, pick the person whose interval has the latest finish time. If no such interval exists (there is no overlap), pick the next interval with the smallest starting time.
4. Continue until the latest finishing time interval is chosen.

Greedy Choice Property: Choosing an interval that lasts longer will minimize the number of intervals, and if there is only one person on shift, it is optimal to have them be designated as the safety contact since whenever anyone is in a room someone has to be the safety contact.

Proof of Correctness: Let S be an optimal set of intervals and G be the greedy set. Assume $|S| > |G|$. By the Greedy Choice Property, we can assume that $g_1 \epsilon S$. Let $A' \subset A$ be the set of intervals that can be started after g_1 starts. By Strong Induction, $G \backslash \{g_1\}$ is optimal for A'. Hence $|G| - 1 \leq |S| - 1$ and so $|G| \geq |S|$, a contradiction.

Runtime: Sorting the list takes $O(nlogn)$ time, and iterating through the list takes $O(n)$ time. The algorithm takes $O(n)$ time.

Problem 3 *(Collaborators: None)*

Part 3(a)

$G = (V, E, w)$

S $T(u)$=the weight of the shortest even path from s to u for all $u \epsilon V$
$D(u)$=the weight of the shortest odd path from s to u for all $u \epsilon V$

R for each $v \epsilon Adj^-(u)$ with count $i = 1, 2, ...|Adj^-(u)|$
$p_i = T(v) + w(v, u)$ if $w(v, u)$ is even
$p_i = D(v) + w(v, u)$ if $w(v, u)$ is odd
T(u)=min$(p_1, p_2...p_n)$ where $n = |Adj^-(u)|$
for each $v \epsilon Adj^-(u)$ with count $i = 1, 2, ...|Adj^-(u)|$
$p_i = D(v) + w(v, u)$ if $w(v, u)$ is even
$p_i = T(v) + w(v, u)$ if $w(v, u)$ is odd
D(u)=min$(p_1, p_2...p_n)$ where $n = |Adj^-(u)|$

B $T(s) = 0$ and $T(u) = D(u) = \infty$ for all vertices u unreachable from s. $D(s) = \infty$

T Topological order of G

O Output is $T(u)$

T $O(|V| + |E|)$ since the runtime is $\sum_{v \epsilon V} O(1 + |Adj^-(v)|)$.

Part 3(b)

$G = (V, E, w)$

S $T(u)$=the weight of the shortest even path from s to u for all $u \epsilon V$
$D(u)$=the weight of the shortest odd path from s to u for all $u \epsilon V$
$x(u)$=the number of optimal even paths from s to u for all $u \epsilon V$

R for each $v \epsilon Adj^-(u)$ with count $i = 1, 2, ...|Adj^-(u)|$
$p_i = T(v) + w(v, u)$ if $w(v, u)$ is even
$p_i = D(v) + w(v, u)$ if $w(v, u)$ is odd

$T(u)$=min$(p_1, p_2...p_n)$ where $n = |Adj^-(u)|$
for each path p_i, count the number r of p_i=min$(p_1, p_2...p_n)$
$x(u) = x(u-1) + r$
for each $v \epsilon Adj^-(u)$ with count $i = 1, 2, ...|Adj^-(u)|$
$p_i = D(v) + w(v, u)$ if $w(v, u)$ is even
$p_i = T(v) + w(v, u)$ if $w(v, u)$ is odd
$D(u)$=min$(p_1, p_2...p_n)$ where $n = |Adj^-(u)|$

B $T(s) = 0$ and $T(u) = D(u) = \infty$ for all vertices u unreachable from s. $D(s) = \infty$. $x(0) = 0$

T Topological order of G

O Output is $x(u)$

T $O(|V| + |E|)$ since the runtime is $\sum_{v \epsilon V} O(1 + |Adj^-(v)|)$.