EECS 340: Assignment 6

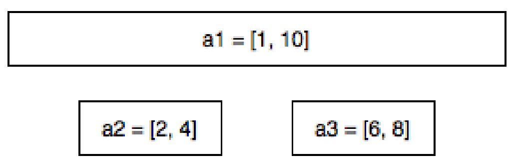
Shaochen (Henry) ZHONG, `sxz517`
Yuhui ZHANG, `yxz2052`

Due and submitted on 04/20/2020
EECS 340, Dr. Koyutürk

Problem 1

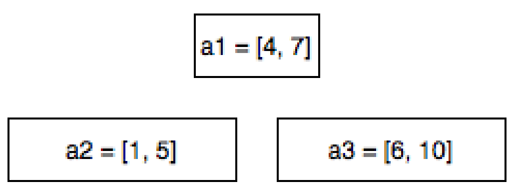
Hi Grader! Please note the boxes in following diagrams are merely provided for visualization purposes, they are not propertionaly graphed. Thanks.

(a)



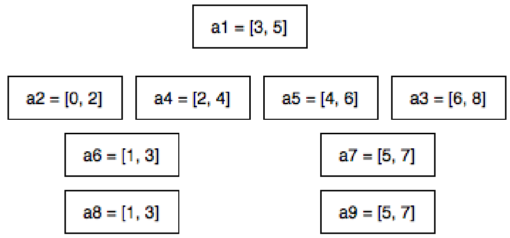
- Greedy Choice: `(a1)`
- Optimal Choice: `(a2, a3)`

(b)



- Greedy Choice: `(a1)`
- Optimal Choice: `(a2, a3)`

(c)



- Greedy Choice: `(a1, a2, a3)`
- Optimal Choice: `(a2, a4, a5, a3)`

Problem 2

Greedy choice property: In some optimal solutions, the activity with the earliest deadline is scheduled first.

Let S be an arbitrarily picked optimal solution, let event E being the activity with the earlist deadline, let event F being the firstly scheduled activity in S. We may have two cases:

- E is the same activity as F, then the property is proven.
- E is not the same activity as F, which means F is scheduled ealier than E. Now we may form a solution S' by swapping the order of E with any activities scheduled before E, until E swaps with F (after such swap E should be the firstly scheduled activity in S).

We may safely perform such swap(s) in the latter case due to the fact that such swap will have no effect on the *maximum delay* time. For any activity i to be scheduled, let Δ_i and Δ_i' represent its delay in solution S and S' respectively. The only case that we will have $\Delta_i' > \Delta_i$ is if activity i is scheduled before E in S (thus after swaps, scheduled after E in S', and caused even more delay). In such scenario we may have:

$$\begin{aligned}
\Delta_i' &= \Delta_i + t_E = f_i - d_i + t_E \\
&\leq s_E + t_E - d_i \\
&= f_E - d_i \\
&\leq f_E - d_e = \Delta_E \\
&\implies \Delta_i' \leq \Delta_E
\end{aligned}$$

Since we know that the maximum delay in S must be $\geq \Delta_E$ (for either E causing the maximum delay or some other activity), and also since Δ_i' is calculated from arbitrarily picked i in S', we may conclude that the maximum delay in S' will not be larger than the maximum delay in S. Consider S is set to be optimal – meaning its maximum maximum delay cannot get any shorter – we must be $\Delta S'_{\max} = \Delta S_{\max}$; which means S' is also optimal. And since E is the firstly scheduled activity in S', we have proven the greedy choice property to be valid.

Problem 3

(a)

Assume we have a set of coins with value of $\{\$1, \$5, \$8\}$ and we are looking to exchange $n = 20$.

The greedy chioce will lead to a solution $\{2 \times \$8, 4 \times \$1\}$ for a totle of 6 coins, however the optimal solution should be $\{4 \times \$5\}$ for total of 4 coins. $4 < 6$ and thus disprove the greedy choice property.

(b)

Assuming we have an optimal solution of $S =< x_o, x_1, x_2, ..., x_k >$, where each x_i respectively represent the amount of c_i coin in the avalivable coin denominations $\{c_1, c_2, ..., c_k\}$ for $1 \leq i \leq k$.

We may observe that there must be $x_i < \frac{c_{i+1}}{c_i}$[1] for all $1 \leq i \leq k$; as otherwise we may have solution $S' =< x'_o, x'_1, x'_2, ..., x'_k >$ with $x'_i = x_i - \frac{c_{i+1}}{c_i}$ and $x'_{i+1} = x_{i+1} + 1$[2]. Due to the fact that $\frac{c_{i+1}}{c_i}$ is at least 2, S' will be a more optimal solution than S and therefore voids the optimal assumption of S. Thus, the property of $x_i < \frac{c_{i+1}}{c_i}$ must be held in the optimal solution.

This means the optimal solution shall have as much as "larger coins" as possible, there will be no scenario that mutiple "smaller coins" can be added to a "larger coin" under an optimal solution.

Knowing this property, we may safely reach the optimal solution S by keep greedily picking the coins with largest c value possible ($c_{\max} \leq n$). Since if we don't pick – or don't pick the largest amount possible of – c_i coin(s) at one point, we will have to pick more c_j coins after (for $1 \leq j < i \leq n$), and that will result in a less optimal solution. Thus, to achieve the optmal solution, we may just enforce the greedy choice at every step.

[1] I wrote it generally, but $\frac{c_{i+1}}{c_i}$ should be a constant 2 in this problem.
[2] This increment might have to be repeatedly done until the condition of $x_i < \frac{c_{i+1}}{c_i}$ is reached.