Problem Set 9 (and last!)

Assigned: July 22
Due: July 29
Name: Subhankari Mishra

Problem 1

Consider the following scheduling problem. Each task I has a length L[I] and a deadline D[I]. There is a single processor that handles one task at a time. Let E [I] be the ending time of I under a given schedule. You get paid an amount D[I] – E[I] if you finish task I early and you get fined an amount E[I] – D[I] if you finish task I late.

There is a very simple greedy algorithm that maximizes your total reward. Find it.

Ans: Below is the greedy algorithm that maximizes your total reward.

Sort the tasks in increasing order of their length and then schedule the tasks in the same order.

Problem 2

In problem 5 of the scheduling notes, give a linear-time algorithm to compute the quantities W[I].

Ans: Below would be the algorithm to compute the quantities W [I].
Let's copy G to G'
Let U be the corresponding node in G for U' in G'.

```
Calculate_W (G, G') {
        If (G' is empty) {
                Return G;
        }
        For (each U' in G' with no outarcs) {
                If (U has any outarcs in G){
                        W' = max (V.W for each U->V in G);
                } else {
                        W' = 0;
                }
                 U.W = W' + U.L
        }
        Delete all U' in G' with no outarcs and their inarcs if any.
        Calculate_W( G,G')
}
```

Problem 3

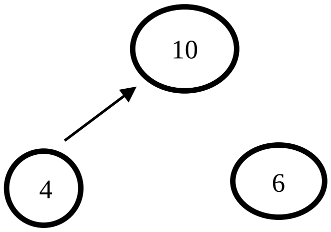
Consider the following scheduling algorithm. All tasks have length 1. There is a single processor. Tasks have a partial ordering represented as a DAG. Each task I has a value V [I]. There is an overall deadline D. The objective is to find the schedule with the maximum possible total value.

A. The obvious greedy algorithm is just to always carry out next the most valuable task that is ready:

```
For (I=0 to D–1) {
    Q = the ready task with the largest value of V[I];
    Schedule Q for time I;
}
```

Construct an example with 3 tasks where this does not find the optimal solution.

Ans:



Let the Deadline be 2.

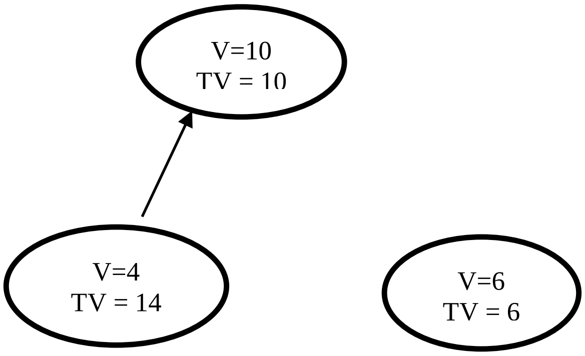
As per the algorithm first task with value 6 will be scheduled and then task with value 4 resulting in a total value of 10.

Whereas if we would have scheduled the task with value 4 first then the task with value 10 we would have got total value of 14 which is greater than 10.

B. Propose an alternative greedy algorithm that will works for the example that you gave as a solution to (A).

Ans: Calculate TV [I] as the value of the maximum possible value from I to DONE.

```
For (I = 0 to D – 1) {
        Q = the ready task with the largest value of TV [I];
        Schedule Q for time I;
}
```

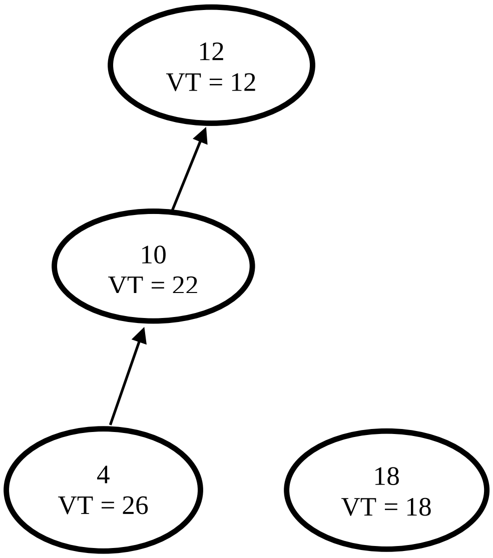


Let the deadline be 2.

As per the above mentioned algorithm the task with value 4 or with TV 14 would be scheduled first and then the task with TV 10, resulting in a total value of 14.

C. Construct an example where the greedy algorithm in (B) does not work

Ans:



Let the Deadline be 2.

As per the greedy algorithm it will schedule the task with VT 26 first and then the task with VT 22, resulting in total value of 14.

Whereas the maximum value would have been 22(4 + 18) if 18 would have been schedules first and then 4.

Problem 4

Modify problem 4 of the notes for the case where each task has its own deadline D [I]. That is: Each task I has length L[I] value V [I], and deadline D[I]. There is a single processor that executes one task at a time. You get paid V [I] if you finish the execution of I before D [I] and 0 if you do not.

A. Propose a greedy strategy for solving this problem.

B. Do an exhaustive search for the optimal solution for the problem below: Use your greedy solution in (A) to begin with a fairly good solution. Use branch and bound to eliminate branches that are worse than the best solution you have already found. Show a trace of the execution of your algorithm.

Task	A	B	C	D	E	F
Length	10	7	5	4	6	5
Value	10	14	9	7	6	5
Deadline	20	8	6	9	15	15

Ans:

A:

Below is the proposed greedy strategy for this problem.

Sort the tasks in decreasing order of value.

Then schedule the task starting with the first task in the sorted list and then moving on down the list and schedule the next job if it fits as per its deadline.

As per this algorithm the above problem would result in a max value of 24 i.e. {B, A}

B: To do the exhaustive search for the optimal solution we will begin by listing the tasks in decreasing order of their value.

Task	B	A	C	D	E	F
Length	7	10	5	4	6	5
Value	14	10	9	7	6	5
Deadline	8	20	6	9	15	15

and then scheduling a task out of the list and identifying the feasible options available after each task is scheduled as per the remaining tasks deadline and current length.

We maintain two arrays In and Out at each step and calculate the total value from the elements in the In set and compare it with the best solution found till that point.
In set will have set of feasible tasks. Out set will have the set of tasks which are not feasible in the current schedule.

The search proceeds as follows:
In {} Out {} Length = 0; Value = 0; Best Solution: 0

Include B. C, D no longer fits.
In {B} Out{C, D} Length = 7; Value = 14; Best Solution: 14

Include A. E, F no longer fit.
In {B, A} Out{C, D, E, F} Length = 17 Value = 24; Best Solution: 24

All tasks decided. Backtrack.

Exclude A.
In {B} Out{C, D, A} Length = 7 Value = 14;

Include E. F no longer fits.
In {B, E} Out{C, D, F} Length = 13 Value = 20;

All tasks decided. Backtrack.

Exclude E.
In {B} Out{C, D, A, E} Length = 7 Value = 14;

Include F.
In {B, F} Out{C, D, A, E} Length = 12 Value = 19;

All tasks decided. Backtrack.
Exclude F.
In {B} Out{C, D, A, E, F} Length = 7 Value = 14;

We have tried all the possible feasible schedules starting with task B.
Let’s move on to the next task in the list as the first task to be schedules which is A.

Include A. B, C, D, E no longer fit.
In {A} Out {B, C, D, E} Length = 10 Value = 10;

Include F.
In {A, F} Out {B, C, D, E} Length = 15 Value = 15;

All tasks decided. Backtrack
Exclude F.
In {A} Out {B, C, D, E, F} Length = 10 Value = 10;

We have tried all the possible feasible schedules starting with task A.
Let’s move on to the next task in the list as the first task to be schedules which is C.

Include C. B no longer fits.
In {C} Out {B} Length = 5 Value = 9;

Include D.
In{C, D} Out {B} Length = 9 Value = 16;

Include E. F, A no longer fit.
In{C, D, E} Out {B, F, A} Length = 15 Value = 22;
All tasks decided. Backtrack.

Exclude E.
In{C, D} Out {B, E} Length = 9 Value = 16;

Include F. A no longer fits.
In{C, D, F} Out {B, E, A} Length = 14 Value = 21;
All tasks decided. Backtrack.

Exclude F.
In{C, D} Out {B, E, F} Length = 9 Value = 16;

Include A.
In{C, D, A} Out {B, E, F} Length = 19 Value = 26; Best Solution: 26
All tasks decided. Backtrack.

Exclude A.
In{C, D} Out {B, E, F, A} Length = 9 Value = 16;
We have tried all the possible feasible schedules starting with task C and then D.

Exclude D.
In {C} Out {B, D} Length = 5 Value = 9;

Include E. F, A no longer fit.
In {C, E} Out {B, D, F, A} Length = 11 Value = 15;
All tasks decided. Backtrack.

Exclude E.
In {C} Out {B, D, E} Length = 5 Value = 9;

Include F.
In {C, F} Out {B, D, E} Length = 10 Value = 14;

Include A.
In {C, F, A} Out {B, D, E} Length = 20 Value = 24;
All tasks decided. Backtrack.

Exclude A.
In {C, F} Out {B, D, E, A} Length = 10 Value = 14;
All tasks decided. Backtrack.

Exclude F.
In {C} Out {B, D, E, A, F} Length = 5 Value = 9;
We have tried all the possible feasible schedules starting with task C.
Let’s move on to the next task in the list as the first task to be schedules which is D.

Include D. B, C no longer fits.
In {D} Out {B, C} Length = 4 Value = 7;

Include E.
In {D, E} Out {B, C} Length = 10 Value = 13;

Include F. A no longer fits.
In {D, E, F} Out {B, C, A} Length = 15 Value = 18;
All tasks decided. Backtrack.

Exclude F.
In {D, E} Out {B, C, F} Length = 10 Value = 13;

Include A.
In {D, E, A} Out {B, C, F} Length = 20 Value = 23;
All tasks decided. Backtrack.

Exclude A.
In {D, E} Out {B, C, F, A} Length = 10 Value = 13;
All tasks decided. Backtrack.

Exclude E.
In {D} Out {B, C, E} Length = 4 Value = 7;

Include F.
In {D, F} Out {B, C, E} Length = 9 Value = 12;

Include A.
In {D, F, A} Out {B, C, E} Length = 19 Value = 22;
All tasks decided. Backtrack.

Exclude A.
In {D, F} Out {B, C, E, A} Length = 9 Value = 12;
All tasks decided. Backtrack.

Exclude F.
In {D} Out {B, C, E, F} Length = 4 Value = 7;

Include A.
In {D, A} Out {B, C, E, F} Length = 14 Value = 17;
All tasks decided. Backtrack.

Exclude A.
In {D} Out {B, C, E, F, A} Length = 4 Value = 7;

We have tried all the possible feasible schedules starting with task D.
Let’s move on to the next task in the list as the first task to be schedules which is E.

Include E. B, C, D no longer fit.
In {E} Out {B, C, D} Length = 6 Value = 6;

Include F. A no longer fits.
In {E, F} Out {B, C, D, A} Length = 11 Value = 11
All tasks decided. Backtrack.

Exclude F.
In {E} Out {B, C, D, F} Length = 6 Value = 6;

Include A.
In {E, A} Out {B, C, D, F} Length = 16 Value = 16;
All tasks decided. Backtrack.

Exclude A.
In {E} Out {B, C, D, F, A} Length = 6 Value = 6;

We have tried all the possible feasible schedules starting with task E.
Let’s move on to the next task in the list as the first task to be schedules which is F.

Include F. B, C no longer fit.
In {F} Out {B, C} Length = 5 Value = 5;

Include A. E, D no longer fit.
In {F, A} Out {B, C, E, D} Length = 15 Value = 15;
All tasks decided. Backtrack.

Exclude A.
In {F} Out {B, C, A} Length = 5 Value = 5;

Include D.
In {F, D} Out {B, C, A} Length = 9 Value = 12;

Include E.
In {F, D, E}. Out {B, C, A} Length = 15 Value = 18;
All tasks decided. Backtrack.

Exclude E.
In {F, D} Out {B, C, A, E} Length = 9 Value = 12;
All tasks decided. Backtrack.

Exclude D.
In {F} Out {B, C, A, D, E} Length = 5 Value = 5;

We have tried all the possible feasible schedules starting with task F.
Search ends.

Best solution found is 26 with schedule {C, D, A}.