Canhong Cai SID:460312484 Assignment1 comp3027

Task1

1. Set $p_i = \min\{1.0, \frac{T}{\sum_{i=1}^{n} t_i}\}$ (i.e. make the same progress on every assignment.)

Suppose there were three assignments, which had the following values:

Assignment(i)	Marks(m_i)	Time(t_i)
1	10	10
2	20	15
3	30	30

.. and we can do T = 40 hours of work before needing a holiday.

If we decided to do $\frac{40}{55}$ (round 72%) of each assignment in this case, we have

- Progress: $p_1 = p_2 = p_3 = \frac{40}{55}$
- Total marks = $p_1m_1+p_2m_2+p_3m_3$ = 43.63
- Total hours worked=$p_1t_1+p_2t_2+p_3t_3$ = 40

This plan requires 40 hours which is equal to T, so it is a feasible solution. However, it is not an optimal solution. In this case, a counter example (optimal solution) is that $p_1 = p_2 = 1$ and $p_3 = 0.5$.

- Total marks = $p_1m_1+p_2m_2+p_3m_3$ = 45
- Total hours worked=$p_1t_1+p_2t_2+p_3t_3$ = 40

2. Sort the assignments in decreasing m_i order. Make as much progress as possible in each assignment, in this order, until T is reached. i.e. if there is T' time remaining when we look at assignment j, then we can set $p_j = \min\{1.0, \frac{T'}{t_j}\}$, and $T' = T' - p_j t_j$.

Suppose there were three assignments, which had the following values:

Assignment(i)	Marks(m_i)	Time(t_i)
1	10	10
2	20	15
3	30	30

.. and we can do T = 40 hours of work before needing a holiday.

If we decided to follow the rule described above, we have

- Progress: $p_3 = 1$, $p_2 = \frac{2}{3}$, $p_1 = 0$
- Total marks = $p_1m_1+p_2m_2+p_3m_3$ = 43.33
- Total hours worked=$p_1t_1+p_2t_2+p_3t_3$ = 40

This plan requires 40 hours which is equal to T, so it is a feasible solution. However, it is not an optimal solution. In this case, a counter example (optimal solution) is that $p_1 = p_2 = 1$ and $p_3 = 0.5$.

- Total marks = $p_1m_1+p_2m_2+p_3m_3$ = 45
- Total hours worked=$p_1t_1+p_2t_2+p_3t_3$ = 40

3. Sort the assignments in increasing t_i order. Make as much progress as possible in each assignment, in this order, until T is reached.

Suppose there were three assignments, which had the following values:

Assignment(i)	Marks(m_i)	Time(t_i)
1	10	10
2	20	15

3	50	30

.. and we can do T = 40 hours of work before needing a holiday.
If we decided to follow the rule described above, we have

- Progress: $p_1 = 1$, $p_2 = 1$, $p_3 = 0.5$
- Total marks = $p_1m_1+p_2m_2+p_3m_3 = 55$
- Total hours worked=$p_1t_1+p_2t_2+p_3t_3 = 40$

This plan requires 40 hours which is equal to T, so it is a feasible solution. However, it is not an optimal solution. In this case, a counter example (optimal solution) is that $p_1 = 0$, $p_2 = \frac{2}{3}$ and $p_3 = 1$

- Total marks = $p_1m_1+p_2m_2+p_3m_3 = 63.3$
- Total hours worked=$p_1t_1+p_2t_2+p_3t_3 = 40$

4. Sort the assignments in increasing t_i/m_i order. For each assignment, in this order, if the assignment can be completed ($p_i = 1.0$), then do so, otherwise do not attempt that assignment ($p_i = 0.0$)

Suppose there were three assignments, which had the following values:

Assignment(i)	Marks(m_i)	Time(t_i)
1	10	9
2	20	15
3	30	30

.. and we can do T = 40 hours of work before needing a holiday.
If we decided to follow the rule described above, we have

- Progress: $p_1 = 1$, $p_2 = 1$, $p_3 = 0$
- Total marks = $p_1m_1+p_2m_2+p_3m_3 = 30$
- Total hours worked=$p_1t_1+p_2t_2+p_3t_3 = 24$

This plan requires 25 hours which is less than T, so it is a feasible solution. However, it is not an optimal solution. In this case, a counter example (Optimal solution) is that $p_1 = p_2 = 1$ and $p_3 = \frac{8}{15}$.

- Total marks = $p_1m_1+p_2m_2+p_3m_3 = 46$
- Total hours worked=$p_1t_1+p_2t_2+p_3t_3 = 40$

5. Sort the assignments in increasing t_i/m_i order. Make as much progress as possible in each assignment, in this order, until T is reached.

Theorem: greedy algorithm described above is optimal.
Proof: (by contradiction)

- Greedy algorithm is not optimal
- Suppose the optimal solution is O and there are two adjacency assignments, say i and j such that $\frac{t_i}{m_i} > \frac{t_j}{m_j}$.
- Building another function O' where we swap the positions of i and j, and leave other assignments in their places.

There are two situations:

1. T allowed all assignments can be finished completely, so i and j can be all finished, which means that process of i and j are 1. Swapping i and j does not change the maximum mark, because no matter doing i first or j first, student can get full mark of these two assignment. Therefore, there is a new optimal solution O' that is

closer to greedy solution. Continuing iteratively until the new optimal solution is transformed in the greedy solution without increasing the cost.
As a result, greedy algorithm must be optimal.

2. T not allowed i and j can be all finished.
Due to the definition of O', mark gained before the last two assignment i and j of O is same as mark gained by O'.
T': remaining time
The difference of Max mark between O' and O: $m_j + \frac{T'-t_j}{t_i} \times m_i - m_i - \frac{T'-t_i}{t_j} \times m_j$
$\rightarrow m_j(1-\frac{T'-t_i}{tj})+m_i(\frac{T'-t_j}{t_i} - 1)$
$\rightarrow \frac{m_j}{t_j}(-T'+t_j+t_i) - \frac{m_i}{t_i}(-T'+t_j+t_i)$
$\rightarrow(-T'+t_j+t_i) \times (\frac{m_j}{t_j} - \frac{m_i}{t_i})$
Because T not allowed i and j can be all completely finished, $(-T'+t_j+t_i)$ must be greater than 0.
Because $\frac{t_i}{m_i} > \frac{t_j}{m_j}$, $(\frac{m_j}{t_j} - \frac{m_i}{t_i})$ must be greater than 0
$\rightarrow(-T'+t_j+t_i) \times (\frac{m_j}{t_j} - \frac{m_i}{t_i})$ must be greater than 0, which means that swapping increases the maximum mark gained.
In conclude, the optimal solution O is not optimal, and greedy solution is optimal.

Task2

Greedy algorithm: If T is equal or greater than half of the time of complete all assignment ($T>=\frac{\sum_{i=1}^{n} t_i}{2}$), it is possible to pass all the assignments and program go on. Firstly, doing all assignments in half of process to make sure that all assignments can be passed. Then sort the assignments in increasing t_i/m_i order. In this order. Make as much as possible in each assignment until T is reached.
Justification: The only difference between the greedy algorithm of task2 and the fifth greedy algorithm of tak1 is the additional constraint ($0.5 \le p_i \le 1.0$). In any possible optimal solution of task2 doing half of process of each assignment is inevitable, which means that it will not affect the output. Therefore, the task2 is quite similar as task1. As we have proven that the fifth solution in task1 is optimal, using the same rule for task2, the result will be still optimal.

Task3

If ($T>=\frac{\sum_{i=1}^{n} t_i}{2}$) → O(1)
then there exists a possible solution
adding up the total mark of doing every assignments in half of process and calculating remaining time (T') after doing all assignments in half of process→ O(n)

<u>Sort</u> assignments by increasing $\frac{t_i}{m_i}$ → O(nlogn)
for each assignment j in the sorting set A { → O(n)
if T'>= half of time of finishing assignment j
MAX=MAX+m_j/2
T' = T' – t_j/2

else

process is $\frac{reaming\ time\ T'}{half\ of\ t_j}$

MAX = MAX + p_j x m_j /2

T' = T' − p_j x $\frac{t_j}{2}$

O(1)+O(n)+O(nlogn)+O(n)=O(nlogn)