1 Deadlines, again!

Solution:

1. The counterexample is shown in table 1. There are three assignment.
   According to the "deadline first" strategy, we run assignment #1 on the first day and get $p_1 = 1$ points. In the next day, whichever we run #2 or #3, we gain extra 5 points so the total is 6.
   However, if we do assignment #2 on the first day and do #3 on the second day, the total points is $p_2 + p_3 = 10$ which is better than the result above. Therefore, the "deadline first" strategy does not always generate the best solution.

i	1	2	3
d_i	1	2	2
p_i	1	5	5

Table 1: Counterexample of the "deadline first" strategy

2. The counterexample is shown in table 2.
   According to the "highest point first" strategy, we run assignment #3 on the first day and #2 on the second day. The assignment #1 will miss its deadline and the total point is $p_2 + p_3 = 7$.
   However, regarding this example, if we exactly do i-th assignment on the i-th day, all of them will be finished and we have 8 points in total. Therefore, this strategy does not always give the best solution either.

i	1	2	3
d_i	1	2	3
p_i	1	2	5

Table 2: Counterexample of the "highest point first" strategy

3. We use k to denote the index of the assignment with the highest points. We claim #k must be contained in any optimal solution S, which results in total point P_S, by contradiction.
   Suppose #k is not done in S, we consider the schedule on d_k-th day in S and there are two cases.
   Case 1: There is no assignment done on d_k-th day. Then, there is a feasible solution S' where we did all the same as that in S except additionally running assignment #k on d_k-th day.
   Note that $P_{S'} = P_S + p_k > P_S$ so S is not the optimal solution. A contradiction occurs.
   Case 2: The assignment #t was done on d_k-th day, $t \neq k$. There is also a feasible solution S' based on S but we do #k on the d_k-th day instead.
   Since $p_k > p_t$ for $\forall t \neq k$ according to the definition of k, the total point resulted from S' is $P_{S'} = P_S - p_t + p_k > P_S$ so S is not the optimal solution. A contradiction occurs.
   Therefore, an optimal solution must contain the assignment with the highest number of points.

4. First, we sort all the assignments in descending order of their points. Next, we arrange the assignments in such order one by one.
   Initially, all days are vacant. When arranging the assignment #i, we look through the current schedule from day d_i-th to day 1 and see if there is any vacant day. If all days have been occupied, we have no time to finish #i but give it up.
   Otherwise, we denote the vacant day as j. If there are multiple ones, we choose the latest day. Then, we schedule #i on day j and mark that day occupied.
   We keep doing this until all assignments are arranged. And the schedule leads to the solution where we can earn the most number of points.

5. We sort the assignments by their points, resulting in D, F, A, B, C, H, G, E, and handle them in order.
   D: Checking from day 5, we schedule D on day 5;
   F: Checking from day 3, we schedule F on day 3;
   A: Checking from day 13, we schedule A on day 13;
   B: Checking from day 2, we schedule B on day 2;
   C: Checking from day 2, day 2 was occupied by B so we schedule C on day 1;
   H: Checking from day 5, day 5 was occupied by D so we schedule H on day 4;
   G: Checking from day 4, day 1-4 were all occupied, we give it up;
   E: Checking from day 7, we schedule E on day 7
   The total number of points earned is $10 + 8 + 7 + 15 + 1 + 12 + 5 = 58$

6. Greedy-choice property: We claim that, given a schedule with certain days vacant, there is always an optimal S solution such that the undone assignment $\#i$ with the most points is run on the latest vacant day before or on d_i.
   We denote that day as t and suppose there is a optimal solution S' where the $\#j$ is done on day t instead. We discuss the case where $i \neq j$; otherwise, we let $S = S'$ and it is already optimal.
   Similarly to that in (3), it is easy to prove that $\#i$ must be contained in S', and we denote its day as s. Next, we swap the schedules on t and s. According to the definition of t, it is safe to put $\#i$ on t. Also, it is safe to move $\#j$ to an earlier day s. We denote the solution after the swap as S. Since $P_S = P_{S'}$, S is optimal and we can always find it with $\#i$ on day t.

7. Optimal-substructure property: We claim that, given a schedule with certain days vacant, after we additionally occupy t by $\#i$, the solution of the remaining assignments on the remaining vacant days are optimal.
   We use A to denote the set of all assignment. Suppose the current optimal solution is S, if the solution T_{A-S} among the remaining vacant days is not the optimal, we can always substitute the solution with the optimal one T'_{A-S} and makes a better new solution $S' = S - T_{A-S} \cup T'_{A-S}$ than the current.
   By contradiction, we deduce the optimal substructure. The key point is that the candidates in the subproblem $A - S$ and the previous decisions S are disjoint so the substitution won't cause conflict.

   By satisfying both the greedy-choice and optimal-substructure properties, this algorithm generates the optimal solution.

2 Don't be late!

We want to get the minimum approximation, which is defined as the time Yan used divided by the time of the optimal strategy in the worst case.
We first see that the optimal strategy is: if the elevator comes before $S - E$ seconds, wait for it; otherwise, use the stairs at time 0. If we use x as the time that the elevator comes, and y as the time used by the optimal strategy, then we have

$$y = \begin{cases} x + E, & \text{if } x \leq S - E \\ S, & \text{if } x > S - E \end{cases} \tag{1}$$

The example in the problem is illustrated in Fig.1.
Now we wait W seconds. So for the first W seconds, our line concides with the optimal line. But after W, our time is a fixed number $W + S$. So our line looks like Fig.2.
The worst-case ratio is when $x = W$, the ratio is $\frac{W+S}{W+E}$. We can write it as $\frac{W+E-E+S}{W+E} = 1 + \frac{S-E}{W+E}$. So larger W gets smaller ratio. But if W becomes larger than $S - E$, the denominator of the ratio is fixed to S, and the numerator is still $W + S$, which leads to larger ratio.
So the best pick is $W = S - E$, leading to the approximation to be $\frac{2S-E}{S}$.

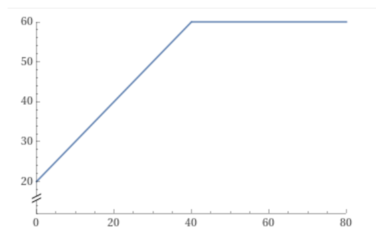

Figure 1: The Optimal Strategy

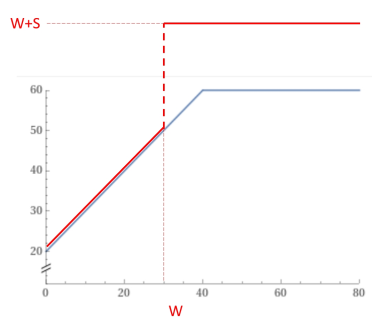

Figure 2: Our time (red line)