Assignment 3 Problem 2

The greedy algorithm stated will produce the happiness of h[3]+h[1]=4

But the optimal result should be h[2]+h[3]+h[1]=6

The greedy algorithm stated will produce the happiness of h[2]+h[1]=5

But the optimal result should be h[1]+h[3]=6

1. Assume an optimal solution contains an trail decided to go at d[i] > n. Since the total number of trials is n. This means that in the first n days, at least one day that has no trial. Therefore, if decide to go trial i at such day, then it is still a optimal solution since the choice of trials is not changed.
2. The algorithm should firstly sort trials based on happiness of each trial in the non-increase order, during sorting algorithm, if we reach two identical happiness, this time, we sort them by the corresponding h[ ] in the non-decrease order. The sorting should take O(n log n) time.

After sorting, we obtain an array of trials with happiness non-increase, and for entries which has the same happiness, the d[ ] of them are in ascent order. And this time, we traverse the trial array in order.

Build another array called isOccupy[ ] which stores bool values and if isOccupy[i] = 0, then this means the day i is available now, otherwise day i is already selected to have a trial. And this array should be set to 0 initially.

While traversing the trial array, for each trial (say current trial structure is x), firstly check isOccupy[x.d], if it is 0, then we choose to go x at day x.d. Otherwise, we traverse isOccupy[ ] from x.d to the beginning of the array. If we meet an value 0, then we choose to go x that day. If we all values are 1, then we will not choose trial x.

The length of isOccupy[ ] should be n, and trial array is also of length n, therefore, the running time of the algorithm is

To prove the correctness, let A = {t1, … ,tn}, be the final output sorted by the same method previously. And B = {b1, … bn} be an optimum solution sorted by the same method.

We want to show that a1, … , a(i-1), bi, …, bn is an optimal solution for all i where 1 < i n. We will induct on i.

Base case: i = 1, since b1 is the has the greatest happiness value, since the way we do greedy algorithm, we guarantee that a1 b1. And a1.d < b1.d (this means that we can also put a1 trial on day b1.d).

Induction step: Suppose a1 … a(i-1) bi … bn is an optimal solution, 1 < i . Since we can also take trial ai on bi.d, and ai.h bi.h, therefore,

a1 … a(i-1) ai b(i+1) … bn is as good as a1 … a(i-1) bi … bn which is an optimal solution. Therefore, the algorithm stays on the optimal path and finally terminate with a optimal solution.