Heat Stroke (Solution)

Author: Hirotaka Yoneda (square1001)

0 Introduction

In this problem, we need to calculate how many patients will be sent by helicopter in the worst case. It should be noted that, compared to problems to "optimize something", which is much more usual in competitive programming, this is problem is to compute "worst case", so we need to think a bit differently.

This problem contains a wide levels of subtasks, which are expected to be tackled by beginners to world-class competitive programmers. Needless to say the importance of partial points in OI-style contests, make sure to get all subtasks that are within your skill level*1.

*1 For example, if you are aiming for IOI Gold Medal, solving subtasks 1-6 (totals 80 points) is a "nice" goal.

Figure 1 The worst scenario for Sample Inputs 1-5 (Number in purple square is the capacities of hospitals; For Sample Input 2, 3, 5, other "worst scenarios" exist). How many of them did you solve?
1 **Subtask 1** \((A_1 \leq A_2 \leq \cdots \leq A_N)\)

When \(A_1 \leq A_2 \leq \cdots \leq A_N\), it is worst to send patients rightward in the case of "both hospitals are open". The reason is that, when all patients of road \(x\) are sent to hospitals, all not-yet-sent patients are in road \(x + 1\) or more right, which means: at this moment, the less vacant the hospital \(x + 1\) is, the worse. Let's simulate this "worst strategy" and count the number of patients sent by helicopter.

2 **Subtask 2** \((L \leq 18, N \leq 18, C_i = 1)\)

We consider brute force, which means to search all possible scenarios of sending patients to hospitals. One method is to brute force all \(2^N\) ways of assignment of, for each patient, "when left/right hospitals are both open, which direction (left or right) to send". Then, we can compute the worst case in \(O(2^N(L + N))\) time, which pass Subtask 2.

3 **Subtask 3** \((L \leq 18, N \leq 100, C_i = 1)\)

When \(L = 18, N = 100\), is there are cases that there are exactly \(2^{100}\) different scenarios? The answer is no. When \(L\) is smaller than \(N\), the number is far less than \(2^N\). Let's explain the reason. When sending a patient, there are 2 choices of how to send, only when left/right hospitals are both open (otherwise there is only 1 choice). Since the total capacity of hospitals in the island is \(L\), the event of "choose from 2 choices" occurs at most \(L - 1\) times. Therefore, at most \(2^{L-1}\) scenarios exist\(^2\).

Using **depth first search (DFS)**, we can brute force all possible scenarios. We can solve this problem in \(O(2^L(L + N))\) time, which pass Subtask 3.

\(^2\) In fact, there is no case to have exactly \(2^{L-1}\) scenarios (when \(L \geq 3\)). We can prove that the maximum number of scenarios is \(O(1.62^L)\). The proof is left for skilled readers.
4 Subtask 4 \((L \leq 100, N \leq 100, C_i = 1)\)

Interpret as an Optimization Problem

The setting of this problem is to "find the worst case", which is rather unusual. However, when we consider the problem like the following, we can think like "Among those that satisfies XX, calculate the maximum value of YY", which is a common pattern in competitive programming problems.

**Interpretation of Problem**  Consider the assignment of the way of sending (left/right/helicopter) for each patient (there are \(3^N\) such ways). Among those that is "a valid scenario", we want to find the maximum number of patients sent by helicopter.

**The Condition of "Validness"**

When we want to solve such problems by Dynamic Programming (DP), we first consider a good characteriza-
tion of "validness of scenario". Figure 2 shows the 3 ways of assignment of left/right/helicopter, but Example 2 and 3 are invalid. Let's consider the reason.

- The violation in Example 2 is that, when hospital 3 is full, a patient on road 2 is sent to there.
- The violation in Example 3 is that, when hospital 3 is open, a patient on road 2 is sent by helicopter.

Let’s define to say that "the violation at hospital 3 is happening". It should be noted that, when checking

---

*3 The cases of violation at hospital \(x\) is (1) when hospital \(x\) is full, a patient on road \(x - 1\) is sent to right (2) when hospital \(x\) is full, a patient on road \(x\) is sent to left (3) when hospital \(x\) is open, a patient on road \(x - 1\), \(x\) is sent by helicopter.
the violation at hospital 3, we don’t use any information on patients of roads 1, 4, 5. To generalize, checking the violation at hospital $x$ only requires the information of (left/right/helicopter for) patients on roads $x - 1$ and $x$.

**Strengthening the Constraints**

In the constraint of $C_i = 1$, when there are 3 or more patients in a single road, the 3rd or later patients are surely sent by helicopter. Thus, erasing all such patients, we can solve the problem with case of at most 2 patients per road. In some competitive programming problems, the technique of "strengthening the constraints" like this works.

**Do Use Dynamic Programming**

Let’s put things together and consider the DP solution. We consider deciding left/right/helicopter from the patients of the leftmost road. Since checking violation at hospital $i$ only requires the information of roads $i - 1$ and $i$, we can consider the following DP:

$$d_{pi}i[F]$$ When left/right/helicopter for patients on roads $1, 2, \ldots, i$ are already decided, the assignment of left/right/helicopter for patients on road $i$ is expressed as $F$, and there are no violation at hospitals $1, 2, \ldots, i$, the maximum number of patients that are already sent by helicopter.

**State Transition** The condition that can be transited from state of $d_{pi}[i - 1][F']$ to state of $d_{pi}[i][F]$ is that, there are no violation at hospital $i$ when the assignment of roads $i - 1, i$ is $F'$, $F$, respectively.

Since there are at most 2 patients per roads, there is at most $3^2 = 9$ ($= O(1)$) states of $F$. Therefore, the DP is done in $O(L)$ time.

(Continued on next page.)
5 Subtasks 5, 6 ($L \leq 600, N \leq 600$)

The DP solution explained Subtask 4 section requires, when $C_i$ is not necessarily 1, requires exponential number of states for $F$. However, we only need much limited information to check violation at hospital $i$. In fact, the following two are enough to record:

- $j$: The number of patients on road $i$ that is sent rightward to hospital $i + 1$.
- $t$: The ID of the last patient when the hospital $i + 1$ became full.*4

We consider the DP that calculate $dp[i][j][t]$, the maximum number of patients sent by helicopter in this state.

The important point is that, $t$ is determined after deciding road $i + 1$, so at the state of $dp[i][j][t]$ we make reservation to decide road $i + 1$ that ”the ID of the last patient when hospital $i + 1$ became full” will match to $t$. Then, the computation of all $dp[i][j][?]$ (using $dp[i - 1][?][?]$) takes $O(N^2Z_i)$ time, where $Z_i$ is the number of patients on road $i$. Since $Z_1 + Z_2 + \cdots + Z_{L-1} = N$, we can solve this problem in $O(N^4)$ time.

However, the condition of state transition from $dp[i - 1][j'][t']$ to $dp[i][j][t]$ is complicated, so now let’s consider how to make implementation easier. Instead of setting $t$ to be the exact time that hospital $i + 1$ became full, consider setting $t$ as ”any time between the time hospital $i + 1$ became full and the time of first helicopter on roads $i, i + 1$” (the exception is when the hospital $i + 1$ is still open at last, then we can set $t = N + 1$).

**Condition of State Transition** Let $b$ the number of patients on roads $i$ among patients $1, 2, \ldots, \max(t, t')$:

1. The number of patients on road $i$ to go rightward $j$ ($0 \leq j \leq Z_i$) is inside the following range:
   - Case $t' < t$: (the number of patients on road $i$ among patients $t' + 1, \ldots, t$) $\leq j \leq b$
   - Case $t \geq t'$: $0 \leq j \leq$ (the number of patients on road $i$ among patients $1, \ldots, t$)

2. The number of patients sent to hospital $i$ is ”correct”:
   - Case $t' \neq N + 1$: $j' + (b - j) = C_i$
   - Case $t' = N + 1$: $j' + (b - j) \leq C_i$

Looking closely to the condition 2., for each $(i, j, t)$, there are only $O(N)$ valid choices of $(j', t')$. Therefore, the computation of $dp[i][?][?]$ (using $dp[i - 1][?][?]$) takes $O(N^2Z_i)$ time, which means we can solve this problem in $O(N^3)$ time.

---

*4 If the hospital $i + 1$ is still open at last, set $t = N + 1$. 

Heat Stroke (Solution)– 5 / 6
6 Subtasks 7, 8

Consider further speeding up the DP solution of Subtask 6. First, as a technique of DP speedup, divide the DP transition into the following 3 cases and solve each cases separately, for not to require "if" condition.

1. Case of $t' < t$
2. Case of $t' \geq t$ ($t' \neq N + 1$)
3. Case of $t' = N + 1$

In fact, for all 3 cases, if we calculate DP values of $(j, t)$ in an appropriate order, we can use cumulative max and calculate $dp[i][j][t]$ in $O(1)$ time. It should be noted that the "direction" of using cumulative max is completely different for cases 1., 2., and 3.

Then, the computation of $dp[i][j][?]$ (using $dp[i - 1][?]$) takes $O(NZi)$ time, which means we can solve this problem in $O(N^2)$ time.

7 Remark

Thank you for reading this review. We uploaded C++ solutions of Subtask 5 and 8.