1 Introduction

Let $\omega \in (0,\infty)^n$ be our vector of times where $\omega_i \in (0,\infty)$ is the time it takes person i to cross the bridge. WLOG[1] we assume $\omega_1 \leq \cdots \leq \omega_n$. Suppose there is only one flashlight. Let $\zeta(\omega; n)$ denote the fastest way to get the n people taking times $\omega_1, \ldots, \omega_n$ across the bridge subject to the following constraints:

(i) Only two people may cross the bridge at any one given moment.

(ii) The flashlight must be used to cross the bridge.

(iii) If people i and j cross together, they cross in time $\min\{\omega_i, \omega_j\}$.

Denote a viable move by m, a sequence of moves by M and the set of all viable sequences $\mathcal{M}$. Let $f(M)$ denote the time taken to complete the sequence of moves M.

To get a feel for the problem, lets compute $\zeta(\omega; 3)$ and $\zeta(\omega; 4)$ for arbitrary ω. For notational ease we start everyone on the left of the bridge and the goal is to get everyone to the right. We call a move that sends people left $\rightarrow$ right a *crossing* and a move right $\rightarrow$ left a *return*. We achieve nothing by doing a solo crossing or a duo return so we can immediately rule out all such moves.

Let us first consider $n = 3$. The first move will be to cross the duo (i, j). Suppose $(i, j) = (1, 2)$. Then, the next move will be $(k, 3)$ where $k < 3$ so $f(M) = \omega_2 + \omega_k + \omega_3$ and we clearly win by taking $k = 1$, giving $\zeta(\omega; 3) \leq \omega_1 + \omega_2 + \omega_3$. Now suppose $(i, j) = (1, 3)$. By an analogous argument we again see the best we can do is $\omega_1 + \omega_2 + \omega_3$. Checking the two cases for $(i, j) = (2, 3)$ we see $\zeta(\omega; 3) = \omega_1 + \omega_2 + \omega_3$.

Things are more interesting already when $n = 4$. We know at some point we'll have to cross the 4^{th} person. Suppose we cross them with the 3^{rd}. The best we can do in this case (exercise: verify this) is by taking $M = ((1, 2), (1), (3, 4), (2), (1, 2))$ or $M = ((1, 2), (2), (3, 4), (1))$ and we deduce $\zeta(\omega; 4) \geq \omega_1 + 3\omega_2 + \omega_4$. If we cross the 4^{th} person with anyone else it is not too hard to see that we win the most by crossing them with person 1 and WLOG we can do this first (exercise: verify both of these facts) to get $\zeta(\omega; 4) \geq 2\omega_1 + \omega_2 + \omega_3 + \omega_4$. We deduce

$$\zeta(\omega; 4) = \min\{2\omega_1 + \omega_2 + \omega_3 + \omega_4, \omega_1 + 3\omega_2 + \omega_4\}$$

In particular, our solution now depends on the structure of ω.

In the $n = 4$ case we saw the emergence of two possible strategies. We can either use person 1 as an escort, pairing them up person $2 \leq i \leq 4$, always using them to return

[1] If ω is not sorted we may do this is $O(n \log n)$ and update our runtimes accordingly.

return the flashlight, or we can pair up our slowest two and use both 1 and 2 as returners. We call these strategies the *escort strategy* and *tag-team strategy* respectively. These will form the basis of our dynamic programming solution.

2 Dynamic Programming Solution

In this section we derive an $O(n)$ algorithm for sorted ω by taking a dynamic programming approach. Our key theorem is as follows.

Theorem 2.1. Fix arbitrary ω, n. Denote the fastest $k \leq n$ people by $\omega|_k := (\omega_1, \ldots, \omega_k)$. Then

$$\zeta(\omega; n) = \zeta(\omega|_{n-2}, n-2) + \min\{2\omega_1 + \omega_{n-1} + \omega_n,\ \omega_1 + 2\omega_2 + \omega_n\}$$

Furthermore, we have an $O(n)$ algorithm for finding both the minimal crossing time and the sequence of moves that gives this time.

To prove this, we need the following lemma. Call a strategy *optimal* if it ends with everyone on the RHS in $\zeta(\omega; n)$ time.

Lemma 2.2. All optimal strategies only use the first two people for return steps.

Proof. Suppose we have a strategy with person $i \geq 3$ returning. Then there must have been an (i, j) move prior for some $1 \leq j \leq n, j \neq i$. If $j < i$ then we can reduce our total time by instead returning j. If $j > i$, then we can instead send across either $(1, j)$ or $(2, j)$ and return person 1 or 2 to reduce our total time. □

Now we're ready to prove our theorem.

Proof of theorem 2.1. By lemma 2.2 we know persons n and $n-1$ won't be used in any return steps and hence WLOG we may focus on crossing those people first and getting the torch back to the LHS, leaving us in the initial state for the problem with $\omega|_{n-2}$. Lemma 2.2 also says our first move must contain either person 1 or person 2. If our first move contains person 2, to be optimal they must be paired with person 1, or we could gain by swapping person 2 with person 1. This gives us two possible strategies for getting the pair $(n-1, n)$ across, taking times $2\omega_1+\omega_{n-1}+\omega_n$ and $\omega_1+2\omega_2+\omega_{n-1}$ and corresponding to the escort / tag-team strategies respectively. To obtain the $O(n)$ algorithm, simply compute this minimum and append either $((1, n), (1), (1, n-1), (1))$ or $((1, 2), (1), (n-1, n), (2))$ to our list of moves iteratively until we have at most 3 people remaining. This remaining time is given in section 1. □

3 Shortest Path Solution

4 Constraint Programming Solution