

CS 577 - Homework 4
Sejal Chauhan, Vinothkumar Siddharth, Mihir Shete

1 Graded written problem

Input: An alphabetical list with the names of all n kayakers for a given day, together with the times they will arrive at the parking lot.

Constraints: A single round-trip lasts m minutes and we can take k kayakers in one trip. The kayakers should be served with FCFS discipline but we can decide for each roundtrip when the bus leaves and how many kayakers we take along.

Output: Organize the schedule so that we are done with our task of dropping the kayakers to the kayak launch platform as early as possible.

1.1 Algorithm

We will solve this problem using a greedy strategy. Our strategy works by minimizing the number of trips and minimizing the starting time of each trip. Our algorithm starts by ordering the list by arrival time (earliest arrival time first). The running time for this initial sort is $O(n \log n)$.

For this ordered list we will employ the following strategy:

1. If there are k_i kayakers in the parking lot at time t_i and there are no more scheduled arrivals in the interval $t \geq t_i$ to $t \leq t_i + m$ then we should carry the k_i kayakers immediately. (if $k_i \geq k$ then look at *Point 3*)
2. If there are $k_i, k_{i+1}, k_{i+2} \dots k_{i+j}$ kayakers arriving at $t_i, t_{i+1}, t_{i+2} \dots t_{i+j}$ and each arrival time is within m minutes of the immediate next arrival time (i.e $t_{i+1} - t_i \leq m$ and so on). Also, $t_i - t_{i-1} > m$ and $t_{i+j+1} - t_{i+j} > m$. Let t be the minimum number of trips required to drop $\sum_{i=i}^{i+j} k_i$ kayakers. We will start our first trip as soon as we have atleast $t \bmod k$ kayakers at the parking lot. And, if $t > k$ the second trip will begin when the number of kayakers waiting at the parking lot are k . This way we are trying to minimize the start time of the first trip keeping the minimum number of trips constant.
if we start the first trip when kayakers are less than $(t \bmod k)$ then we will need more than t trips and hence more time, but if we start our first trip later than when $(t \bmod k)$ kayakers arrive, our last trip will end a little later.
3. If at any point of time there are $\geq k$ kayakers at the parking lot then we start the trip immediately with k kayakers.