

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 round trip 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 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:

- We will start looking at the number of kayakers arriving at each time from the end of our sorted list. And from the end of the list we start making groups of k kayakers. We will go on doing this grouping till the number of kayakers remaining are $n \bmod k$.
- We will start our first trip as soon as we have at least $n \bmod k$ (or k if $n \bmod k = 0$) kayakers at the parking lot. The second trip and all the trips after that will begin when the number of kayakers waiting at the parking lot are k . This way we are trying to minimize the number of trips and also minimize the wait time for every k^{th} person after the first trip.

1.2 Greedy Stays Ahead Proof

Our greedy strategy is to accommodate as many kayakers on a trip from the last arrival time so that the trips taken to transport the kayakers are minimized and the wait time for every k^{th} person after the 1st trip is minimized (if $n \bmod k = 0$, then the wait time for every k^{th} person is minimized).

We will be using strong induction to prove that the waiting time every k^{th} person after the 1st trip is minimized.

Base Case:

The base case will vary according to the number of kayakers n coming in on the day.

1. n is a factor of k :

In this case we will start the **first trip** as soon as the k^{th} kayaker arrives, so the waiting time for this person is 0 and the waiting time cannot get any better than this.

2. n is not a factor of k :

In this case our greedy algorithm will start the **first trip** when we have $n \bmod k$ kayakers. If the round trip time m is such that the bus arrives before the k^{th} kayaker from the first trip then it is trivial to see that this kayaker will not have to wait at all.

Now, in the case when m is such that the k^{th} kayaker arrives before the bus returns from the first trip then that kayaker will have to wait for some time. And we will prove that this is the minimum wait time which is possible.

To prove our claim, let's assume that in the first trip the bus carries some kayakers less than $n \bmod k$ kayakers. So, the number of kayakers left after the first trip will be more than k and so the bus will have to make 1 extra trip before picking up the k^{th} kayaker and so he will have to wait for more time than our greedy algorithm.

Next, let's assume that in the first trip the bus carries some kayakers more than $n \bmod k$. Then, the bus can carry the k^{th} kayaker in the next trip itself, but this second trip will start later than the second trip of our greedy algorithm and the kayaker will have to wait longer.

From the above 2 arguments we can see that the greedy algorithm will minimize the wait time for the k^{th} kayaker after the first trip.

Inductive Step:

Assume that the waiting time for k^{th} kayaker in i^{th} trip is minimum possible waiting time. Then, if the round trip time m is such that the k^{th} kayaker in $(i + 1)^{th}$ trip arrives at the parking lot before the bus returns from the i^{th} trip then this kayaker will have to wait. Since the round trip time for bus is constant and the wait time for the k^{th} person in the last trip was minimum it trivially follows that the wait time for the k^{th} person in the $(i + 1)^{th}$ trip is minimum. Also, it is trivial to see that if the bus comes before the k^{th} kayaker in $(i + 1)^{th}$ trip then his waiting time will be 0.

Conclusion:

We have proved by induction that the waiting time for k^{th} person in every trip is minimal, and every trip after the first one always has k kayakers, which implies that a trip will end in the least possible time since the round trip time for the bus is a constant value.

1.3 Time Complexity

The initial sorting operation takes $O(n \log(n))$ time and traversing the list and making groups of people will be linear in n . So the overall time complexity of our algorithm will be $O(n \log(n))$