CS217 Data Structures and Algorithm Analysis
Fall term 2025

Exercise Sheet 12

Handout: November 25th — Deadline: December 2nd, 4pm

Question 12.1 (0.5 marks)

Suppose that you modify GREEDY-ACTIVITY-SELECTOR to use the following greedy strategies. State whether each strategy would yield an optimal solution or not. If they do, then provide a proof of optimality. If they don't, then provide an example instance where the strategy fails.

1. Always select the activity of least duration amongst those that are compatible with all previously selected activities
2. Always select the compatible activity that overlaps with the fewest remaining activities
3. Always select the last activity to start that is compatible with all previously selected activities
4. Always select the compatible activity with the earliest start time

Answer

1. This strategy does not yield an optimal solution. Consider the following set of activities with their start and finish times:

i	1	2	3	4
s	1	3	0	4
f	4	5	6	7
d	3	2	6	3

   Using the least duration strategy, we would first select Activity 2, then we cannot select any other activities because they all overlap with 2. However, the optimal solution is to select Activities 1, 4, which gives us a total of 2 activities.
2. This strategy does not yield an optimal solution. Consider the following set of activities with their start and finish times:

i	1	2	3	4	5	6	7	8
s	0	0	0	1	4	5	7	8
f	1	3	4	5	6	8	9	10
n	2	3	3	3	2	2	2	1

```
-1
---2
----3
 ----4
    --5
     ---6
       --7
        --8
```

Using the fewest overlaps strategy, we would first select Activity 8, then we can select 1 and 5, which gives us a total of 3 activities. However, the optimal solution is to select Activities 1, 4, 6, and 8, which gives us a total of 4 activities.

This strategy looks only at how many intervals an activity overlaps now, not at how that choice affects the possibility to chain many compatible activities later. An interval might have very few overlaps (so it looks “safe”), but selecting it can prevent constructing a long chain of short, back-to-back activities that together yield a larger cardinality.

3. This strategy yields an optimal solution. The proof is similar to the standard greedy activity selection proof. By selecting the last activity to start, we ensure that we leave as much room as possible for the remaining activities. If there was a better solution that included an earlier starting activity, we could replace it with the last starting activity without reducing the number of activities selected.
4. This strategy does not yield an optimal solution. Consider the following set of activities with their start and finish times:

i	1	2	3
s	1	5	0
f	4	7	7

Using the earliest start time strategy, we would first select Activity 3, then we cannot select any other activities because they all overlap with 3. However, the optimal solution is to select Activities 1 and 2, which gives us a total of 2 activities.

Question 12.2 (0.25 marks)

Prove that the fractional knapsack problem has the greedy choice property, hence always finds an optimal solution.

Answer

We have items $1, \ldots, n$, each with value v_i and weight w_i.
We can take a fraction $x_i \in [0, 1]$ of each item, with knapsack capacity W.
Total value: $\sum_{i=1}^{n} x_i v_i$, subject to $\sum_{i=1}^{n} x_i w_i \leq W$.

Define ratio $r_i = \frac{v_i}{w_i}$. Assume we reindex so that
$r_1 \geq r_2 \geq \cdots \geq r_n$.

The greedy algorithm:

1. Sort items so $r_1 \geq r_2 \geq \cdots \geq r_n$.
2. Starting from $i = 1$, take as much as possible of item i:
   - If there is remaining capacity C and $w_i \leq C$: take full item ($x_i = 1$).
   - Else: take fraction $x_i = C/w_i$ of item i and stop (knapsack is full).

Greedy choice property

We want to show: **there exists an optimal solution that takes as much as possible of the item with highest ratio (item 1)**. Then we can repeat this argument for the subproblem.

Let G be the greedy solution, and let O be an optimal solution (not necessarily greedy).

Focus on item 1 (highest ratio r_1). If O already takes the same amount of item 1 as the greedy solution (either completely or the same fraction), we are fine; just look at remaining capacity as a subproblem on items $2, \ldots, n$.

Otherwise, uppose O does not take item 1 “as much as possible” while the greedy solution does. Let x_1^G be the amount of item 1 in the greedy solution, x_1^O the amount in O, and $x_1^O < x_1^G$. Since knapspack is full in both cases, there is some weight in O “assigned” to items with ratio $\leq r_1$ that the greedy solution instead uses for item 1 (ratio r_1). Pick any item j in O with $x_j^O > 0$ and $r_j \leq r_1$. Move a small amount of weight Δ from item j to item 1.
Capacity stays the same, but value may change:

$$\Delta V = \Delta \cdot r_1 - \Delta \cdot r_j = \Delta(r_1 - r_j) \geq 0$$

since $r_1 \geq r_j$.
So this swap does not decrease (and may increase) the total value. We can swap until item 1 is taken by O as much as in the greedy solution. Thus, there exists an optimal solution O' that matches the greedy amount of item 1.

This proves the greedy choice property: **there is an optimal solution that includes the greedy choice (taking as much as possible of the highest-ratio item).**

Optimal substructure

Once we prove that greedy choice is save for the first item, the remaining problem is:

- A knapsack with reduced capacity $W' = W - x_1^G w_1$,
- Remaining items $2, \ldots, n$.

This is again a fractional knapsack problem of the same kind.
By applying the same argument recursively, greedy will be optimal on the remaining items. Thus, by induction, the greedy algorithm constructs an optimal solution for the whole problem.

Therefore, the fractional knapsack problem has the greedy choice property and the greedy algorithm always finds an optimal solution.

Question 12.3 (0.5 marks)

Eddy takes part in a cycle race from start s_1 to finish s_n with feed stations $s_2, \ldots, s_{n-1}$ along the way and distances d_i between s_i and s_{i+1}. To save time, Eddy plans to stop at the smallest possible number of stations. He knows that he can cycle distance ℓ without stopping for supplies, where $\ell > d_i$ for all $1 \leq i \leq n-1$.

(a) Design a greedy algorithm that computes the minimal number of stops for Eddy.
(b) Argue why your greedy strategy yields an optimal solution.

Answer

(a) **Greedy Algorithm:**

1. Initialize an empty list of stops.
2. Set the current position to the start station s_1.

3. While the current position is not the finish station s_n:
    - Find the farthest station s_j such that the distance from the current position to s_j is less than or equal to ℓ.
    - Add station s_j to the list of stops.
    - Update the current position to s_j.
4. Return the list of stops.

(b) **Proof of Optimality:**

Let G be the set of stops chosen by the greedy algorithm, and let O be an optimal set of stops.

We first focus on the first stop. The greedy choice is to stop at the farthest station s_j reachable from the current position. If the greedy choice is also in the optimal solution O, we can proceed to the next stop. Otherwise, if the optimal solution O chooses a different station s_k (where $k < j$), we can replace s_k with s_j in O without increasing the number of stops, because this replacement does not make later stops unreachable.
So there exists an optimal solution that includes the first greedy choice, and we can repeat this argument for the remaining stations. Thus, by induction, the greedy algorithm yields an optimal solution with the minimal number of stops.