S. KIMMEL

Goals

- Practice Dynamic Programming Set-up
- Describe multi-exchange arguments

Test Info

- All topics up to 3/5 except QuickSort
- All PSets up to 4
- Tuesday: review. Topics → Canvas Discussion until Sunday
- Wed @ noon → Frid @ 6 pm
- Reserve 3 hours
- No class 3/14
- 1 page, single sided handwritten notes

Quiz on Canvas

No PSet 5

S. KIMMEL

Scheduling Goal:

Minimize $\sum_{j=1}^{n} w_j C_j$ ⇐ "Objective function" A

Greedy strategy: Order by largest w_i / t_i

Q: If we get rid of assumption that all w_i / t_i are unique, what changes? Which change(s) make the proof fail?

S. KIMMEL

Scheduling Goal:

Minimize $\sum_{j=1}^{n} w_j C_j$ ⇐ "Objective function" A

Greedy strategy: Order by largest w_i / t_i

Q: If we get rid of assumption that all w_i/t_i are unique, why does our exchange argument fail?

A There might not be a unique solution

B) Objective function might not decrease from σ^* to $\sigma^{*'}$

C) We can't create an ordering such that $w_1/t_1 > w_2/t_2 > \ldots > w_n/t_n$

All are true, but B ruins the proof.

Recall, we get a contradiction b/c $\sigma^* = (\ldots, k, j, \ldots)$

where $j < k \Rightarrow \frac{w_j}{t_j} > \frac{w_k}{t_k} \Rightarrow w_j t_k > w_k t_j$

$$\Rightarrow A_{\sigma^*} - A_{\sigma^{*'}} = w_j t_k - w_k t_j > 0$$

Now, if w_i/t_i not unique:

$$j<k \Rightarrow \frac{w_j}{t_j} \geq \frac{w_k}{t_k} \Rightarrow w_j t_k \geq w_k t_j$$

$$\Rightarrow A_{\sigma^*} - A_{\sigma^{*\prime}} = w_j t_k - w_k t_j \geq 0$$

could be equal... not a contradiction

New Idea:
Still use EXCHANGE argument, but now need more exchanges.

Choose some ordering s.t.

$$w_1/t_1 \geq w_2/t_2 \geq w_3/t_3 \cdots \geq w_n/t_n$$

Let σ be strategy using this ordering

Let σ^* be any other strategy

We will show $A_{\sigma^*} \geq A_\sigma$, which means σ is optimal

$\sigma^* = (\ldots k, j \ldots)$ where $w_j/t_j \geq w_k/t_k$

exchange j,k

$\sigma^{*\prime} = (\ldots k', j' \ldots)$ where $\frac{w_{j'}}{t_{j'}} \geq \frac{w_{k'}}{t_{k'}}$

exchange j',k'

$\sigma^{*\prime\prime}$ $\rightarrow$ σ

This procedure is bubble sort!

$$A_{\sigma^*} \geq A_{\sigma^{*\prime}} > A_{\sigma^{*\prime\prime}} \geq A_\sigma$$

Conclusion: $A_{\sigma^*} \geq A_{\sigma}$, so greedy strategy is optimal!

Q What is the runtime of this greedy algorithm?

A) $O(1)$ B) $O(n)$ C) $O(n \log n)$ D) $O(n^2)$

Conclusion: $A_{\sigma^*} \geq A_{\sigma}$, so greedy strategy is optimal!

Q What is the runtime of this greedy algorithm?

A) $O(1)$ B) $O(n)$ C) $O(n \log n)$ D) $O(n^2)$

now some might have the same value.

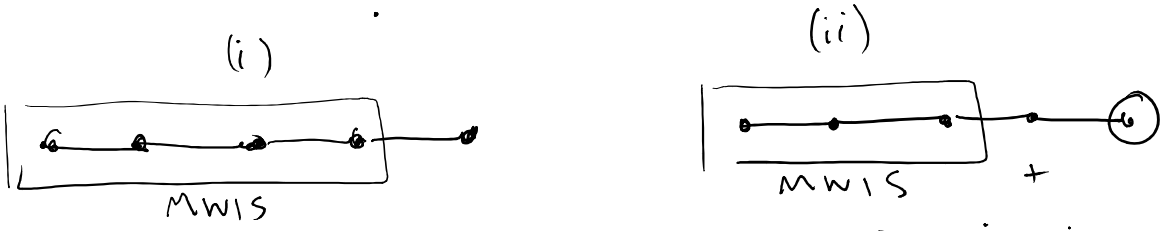
$O(n \log n)$. Need to sort n items, requires $O(n \log n)$ time, bubble sort is only imagined to happen as part of proof. It doesn't actually occur.

S. KIMMEL

To Create D. P. (dynamic programming) algorithm:

1. Think of form of optimal solution.
   - WMIS on line: $v_n \in S$ (ii) or $n \notin S_n$ (i)

2. Optimal soln in terms of smaller soln? (For each case)



3. Create recurence for objective function

$$A[k] = \max\{A[k-1], A[k-2] + w_k\}$$

(i) (ii)

$A[k]$: weight of MWIS on G_k

4. Store A in an array using for-loop

5. Work backwards through array to reconstruct optimal solution

S. KIMMEL

Dynamic Programming - More Practice

Knapsack Problem $K(v_1, \ldots v_n, w_1, \ldots w_n, W)$

Input:
- n items, each has
  - value v_i
  - size w_i

  $\} \in \mathbb{Z}^+$
- Capacity W

Output: A subset $S \subseteq \{1, 2, \ldots, n\}$ that maximizes $\sum_{i \in S} v_i = V(S)$ and satisfies $W(S) = \sum_{i \in S} w_i \leq W$

$W(S) \leq W$: constraints

$\sum_{i \in S} v_i = V(S)$: objective function

Applications

- Cargo trucks
- Investments
  - v_i = expected yield %
  - w_i = min investment amount
  - W = amount to invest

Notation

- Let $K_{i,R}$ be subproblem on 1st i items, with capacity R.
- Solution: satisfies constraints
- Optimal solution: satisfies constraints and maximizes value.

S. KIMMEL

1. Form of optimal solution:

Let S be the optimal solution to $K_{n,w}$

Two options: (i) $n \notin S$

(ii) $n \in S$

2. Relate to optimal solution of subproblem

(i) If S is optimal solution to $K_{n,w}$ and $n \notin S$, then ____ is optimal solution to $K_{-,-}$

Pf: For contradiction assume ___ is not optimal solution to $K_{-,-}$ .....

(ii) If S is optimal solution to $K_{n,w}$ and $n \in S$ then ____ is optimal solution to $K_{-,-}$.

Pf: For contradiction assume ___ is not optimal solution to $K_{-,-}$ .....

S. KIMMEL

(i) If S is the optimal soln to $K_{n,w}$ and $n \notin S$, then S is optimal soln to $K_{n-1,w}$

Pf: For contradiction, suppose S is not opt soln for $K_{n-1,w}$. Then there exists an optimal solution S' for $K_{n-1,w}$ with $V(S') > V(S)$. But S' is also a solution to $K_{n,w}$ with $V(S') > V(S)$ and $n \notin S'$, so S is not the optimal solution, a contradiction

(ii) If S is opt. soln for $K_{n,w}$ and $n \in S$, then $S - \{n\}$ is opt soln for $K_{n-1,w-w_n}$.

Pf: Suppose for contradiction $S - \{n\}$ is not optimal for $K_{n-1,w-w_n}$. Then there exists an optimal solution S' for $K_{n-1,w-w_n}$ s.t. $V(S') > V(S - \{n\})$. But then $S' + \{n\}$ is a soln to $K_{n,w}$, and $V(S' + \{n\}) > V(S)$, so S is not opt for $K_{n,w}$, a contradiction