REAL KNAPSACK PROBLEM

INPUT: $O_1, O_2, \ldots, O_n$ KNAPSACK OF CAPACITY m.

PROFIT	p_1	p_2	...	p_n
WEIGHT	w_1	w_2		w_n

Output: $x_1, x_2, \ldots, x_n$

S.T. $\sum_{i=1}^{n} w_i x_i \leq m$ and $\sum_{i=1}^{n} p_i x_i$ IS MAXIMUM

CSE 350B

Exam 1 STAT

HIGH: 98

LOW: 35

μ: 66.4

σ: 16.2

1,5,6 → ABDEL

2,3,4 → RAJ

Example:

	1	2	3
PROFITS	5	20	50
WEIGHTS	5	10	50

$m = 12$

SELECTION CRITERION 1:

NON INCREASING ORDER OF PROFITS.

$x_3 = \frac{15}{50}$; PROFIT $= \frac{15}{50} \cdot 50 = 15$.

SELECTION CRITERION 2:

NON DECREASING ORDER OF WEIGHTS

$x_1 = 1.0$; $x_2 = \frac{7}{10}$; $x_3 = 0.0$;

PROFIT $= 5 + \frac{7}{10} \cdot 20 = 19$

SELECTION CRITERION 3:

NON INCREASING ORDER OF $\frac{P}{W}$ RATIOS (PROFIT DENSITIES)

$x_2 = 1.0; x_1 = \frac{2}{5}; x_3 = 0.0$

$\text{PROFIT} = 1.0 \times 20 + \frac{2}{5} \cdot 5$

$= 22$

THEOREM: The greedy alg. is optimal when we use CRITERION 3.

PROOF SUMMARY:

Let $x_1, x_2, \ldots, x_n$ be the greedy solution.

Let $y_1, y_2, \ldots, y_m$ be an optimal solution

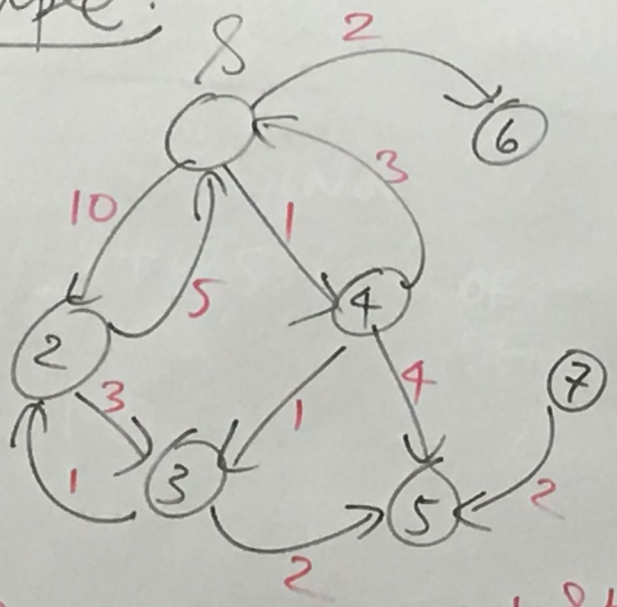
If these two are not identical,
let k be the least index
s.t. $x_k \neq y_k$.

SINGLE SOURCE SHORTEST PATHS PROBLEM.

INPUT: A WEIGHTED DIRECTED GRAPH $G(V, E)$;
A SOURCE $s \in V$

OUTPUT: The SHORTEST PATH FROM s to u for every $u \in V - \{s\}$.

Example:



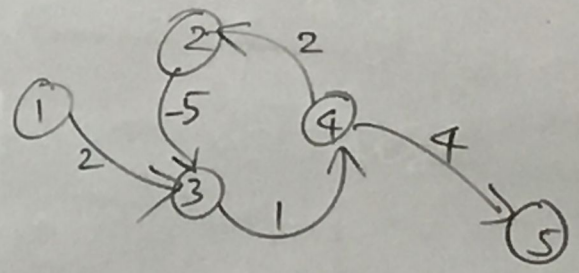
s to 2: s to 4 to 3 to 2 : 3

s to 3: s to 4 to 3 : 2

s to 4 : 1

s to 5 : s to 4 to 3 to 5 : 4

s to 6 : 2 ; s to 7 : ∞



WE ASSUME that there are NO Negative cycles.

CSE 3500 Algorithms and Complexity – Fall 2016
Lecture 16: October 20, 2016

Greedy Algorithms Continued

The first part of this lecture was spent on solving Exam 1 problems. In the second part we continued our discussion on greedy algorithms. Specifically, we talked about the fractional (or real) knapsack problem.

Real knapsack problem (RKP)

- Input for this problem are n objects where each object has a profit and weight. p_i and w_i are the profit and the weight of object i, for $1 \leq i \leq n$. We are given a knapsack of capacity m. The problem is to determine what fraction of each object should be stored in the knapsack such that the capacity of the knapsack is not exceeded and the profit obtained is maximum. More specifically, we want to deterime $x_1, x_2, \ldots, x_n$ such that $\sum_{i=1}^{n} w_i x_i \leq m$, $\sum_{i=1}^{n} p_i x_i$ is maximum, and $0 \leq x_i \leq 1$ for all $i,\ 1 \leq i \leq n$.

- We can employ the greedy approach to solve this problem. There are many ways in which we can order the objects and as a consequence many different selection criteria are possible. We'll investigate some possible criteria using an example.

- Consider an instance of the RKP with three objects whose profits are $5, 20, 30$ and whose weights are $5, 10$, and 30, respectively. The knapsack capacity is 12.

- **Selection Criterion 1:** As per this criterion, the objects are examined in non-decreasing order of the weights. Under this criterion, the first object examined will be 1 whose capacity is 5. We include this object into the knapsack in full and set $x_1 = 1.0$. We then examine object 2. The knapsack has only a capacity of 7 left and hence we decide to include $\frac{7}{10}$ fraction of the second object into the knapsack and set $x_2 = \frac{7}{10}$. Since the knapsack is full now, we set $x_3 = 0.0$. The resultant profit is $1.0 \times 5 + \frac{7}{10} \times 20 + 0.0 \times 30 = 19$.

- **Selection Criterion 2:** This criteion orders the objects in nonincreasing order of the profits. Object 3 is examined first and we set $x_3 = \frac{12}{30}$. Since the knapsack is full with this decision, we set $x_1 = x_2 = 0.0$. The resultant profit is $0.0 \times 5 + 0.0 \times 20 + \frac{12}{30}30 = 12$.

- **Selection Criterion 3:** Here we order the objects based on the profit to weight ratios (called *profit densities*) in nonincreasing order. Based on this criterion, the first object to be examined wil be 2 and we set $x_2 = 1.0$. A capacity of 2 is left. Objects 1 and 3 have the same profit density of 1 and hence we can either choose object 1 or object 3 next. If we go with object 1, we can set $x_1 = \frac{2}{5}$ and $x_3 = 0.0$. The net profit for this solution is $\frac{2}{5} \times 5 + 1.0 \times 20 + 0.0 \times 30 = 22$.

- **Theorem:** The greedy algorithm for RKP will always output the optimal solution under Selection Criterion 3.

  Proof sketch: Assume that the objects have been ordered based on the profit densities in nonincreasing order. Specifically, object 1 has the largest profit density, object 2 has the next largest profit density, etc. Let $y_1, y_2, \ldots, y_n$ be any optimal solution and let $x_1, x_2, \ldots, x_n$ be the greedy solution.

  Note that the greedy solution will be such that $x_1 = 1, x_2 = 1, \cdots, x_{j-1} = 1, x_j < 1, x_{j+1} = x_{j+2} = \cdots = x_n = 0.0$, for some $j \leq n$. If the optimal and greedy solutions are identical, then the greedy solution is also optimal. If not, let k be the least index such that $x_k \neq y_k$. For this value of k, we can show that $x_k > y_k$.

  Change the value of y_k to match x_k. This is done by decreasing the values of $y_{k+1}, y_{k+2}, \ldots, y_n$ as needed. If $z_1, z_2, \ldots, z_n$ is the resultant new solution, then we will have: $z_k = x_k$ and $w_k(z_k - y_k) = \sum_{i=k+1}^{n}(y_i - z_i)w_i$.

  We show that the profit for the new solution $z_1, z_2, \ldots, z_n$ is no less than that of the old solution $y_1, y_2, \ldots, y_n$. We then look for the next least index k' (greater than k) for which $z_{k'} \neq x_{k'}$ and continue the process of matching $z_{k'}$ with $x_{k'}$ and prove that the resultant solution has a profit that is no less than that of the solution $z_1, z_2, \ldots, z_n$.

  Proceeding in this manner, ultimately, we will end up with a new solution where each value of the greedy solution has been matched. Even for this solution we will able to prove that the profit is no less than that of $y_1, y_2, \ldots, y_n$ implying that the greedy solution is also optimal. □

Single source shortest paths (SSSP) problem

- Input: A weighted directed graph $G(V, E)$ and a node $s \in V$ called the source. Output: the shortest paths from s to every other node in G.

- In any path finding problem in graphs, we typically assume that there are no negative cycles (i.e., cycles whose total weights are negative) since otherwise the notion of the shortest path may not be well defined.

- Dijkstra's algorithm for the SSSP problem is greedy and it assumes that there are no negative edges in the graph.

- The algorithm generates the shortest paths in nondecreasing order of the shortest path weights. I.e., the shortest path whose path weight is the least (from across all the shortest paths) is generated first; the shortest path whose path weight is the next smallest is generated next; and so on.

- Dijkstra's algorithm is similar to Prim's algorithm for the MST problem. Similar to the *near* data structure for the MST problem, Dijkstra's algorithm defines a data structure called *dist*.
- More details of the algorithm will be given in the next lecture.