Practice problems

Greedy
Greedy (optimal substructure?)

We have learned in class that the shortest-path problem exhibits optimal substructure (i.e., fragments of a shortest path are shortest paths). Is the same true for the longest simple path? Recall that a path is simple if none of the vertices in the path are repeated.

If you think that the longest simple path problem has the optimal substructure, give a convincing argument (the more formal, the better). If you think that the longest simple path problem does not have the optimal substructure, show an example.

Single source shortest path

Dijkstra’s single-source shortest-path uses a set $C$ (the “closed”) of vertices that initially contains only the source $s$ and that eventually includes all the vertices of the graph $(V,E)$. Vertices are added to $C$ one at a time, as explained in class. Let $f(v)$ the number of times that the label $D[v]$ of a vertex $v$ in $V – C$ changes due to an edge relaxation. Answer each of the following questions (provide a intuitive explanation for each of your answers)

- Can the label $D[v]$ of a vertex $v$ in $V – C$ ever get smaller than the cost of a shortest $s$-to-$v$ path in the graph $G$?
- Can $f(v)$ become greater than the degree of $v$? (recall that the degree of a node is the number of edges incident to it)
- Can $f(v)$ be less than the degree of $v$?
Problem (greedy-choice proof)

Assume that you are given two unsorted arrays \( A = \{a_1, a_2, \ldots, a_n\} \) and \( B = \{b_1, b_2, \ldots, b_n\} \) of \( n \) positive integers. We want to determine an ordering of the elements of \( A \) and \( B \) such that \( W = \prod_{i=1}^{n} a_i^b \) is maximized. Consider the following greedy algorithm.

**Algorithm** **Greedy** \((A, B)\); sort \( A \) and \( B \) in decreasing order; return \((A, B)\)

Show that the greedy-choice property holds for the algorithm **Greedy** (no need to prove that the problem has the optimal substructure property).
Dynamic programming (design)

- Give an efficient algorithm for the following problem
- Given a rod of length $n$, and a set of prices $p_i$, $i=1,2,\ldots,n$ for a rod piece of length $i$, determine the maximum revenue obtainable by cutting up the rod and selling the individual pieces

<table>
<thead>
<tr>
<th>Length $i$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Price $p_i$</td>
<td>1</td>
<td>5</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>17</td>
<td>17</td>
<td>20</td>
<td>24</td>
<td>30</td>
</tr>
</tbody>
</table>
Dynamic programming

Consider a weighted directed graph $G = (V, E)$ where weights are positive integers. Given nodes $s \in V$, $t \in V$ and integer $k > 0$ design a dynamic programming algorithm that determines whether there is a path from $s$ to $t$ of weight exactly $k$. The path does not have to be simple, i.e., vertices can be used more than once. Analyze the time- and space-complexity of your algorithm.