Greedy Algorithms – Exercise Problems

Lecturer: Shi Li

Department of Computer Science and Engineering
University at Buffalo
Maximum Independent Set on Trees

Given a tree $T = (V, E)$, find the maximum independent set of the tree. For example, maximum independent set of the tree of following tree has size 7.

Figure: The green vertices shows that the maximum independent set of the tree has size 7.

Design an efficient greedy algorithm to solve the problem.
Scheduling to Minimize Weighted Completion Time

**Input:** A set of \( n \) jobs \([n] := \{1, 2, 3, \cdots, n\}\)
each job \( j \) has a weight \( w_j \) and processing time \( t_j \)

**Output:** an ordering of jobs so as to minimize the total weighted completion time of jobs

\[
\begin{align*}
t_1 &= 1 & w_1 &= 2 \\
t_2 &= 2 & w_2 &= 5 \\
t_3 &= 3 & w_3 &= 7 \\
\end{align*}
\]

\[
\begin{align*}
w_1 &= 2 & w_2 &= 5 & w_3 &= 7 \\
1 & & 2 & & 3 \\
\end{align*}
\]

\[
\begin{align*}
0 & & 1 & & 2 & & 3 & & 4 & & 5 & & 6 \\
cost &= 2 \times 1 + 5 \times 3 + 7 \times 6 = 59
\end{align*}
\]

\[
\begin{align*}
w_2 &= 5 & w_3 &= 7 & w_1 &= 2 \\
2 & & 3 & & 1 \\
\end{align*}
\]

\[
\begin{align*}
0 & & 1 & & 2 & & 3 & & 4 & & 5 & & 6 \\
cost &= 5 \times 2 + 7 \times 5 + 2 \times 6 = 57
\end{align*}
\]
Driving from $A$ to $B$ using with minimum number of gas stops

You wish to drive from point $A$ to point $B$ along a highway minimizing the time that you are stopped for gas. You are told beforehand the capacity number $L$ of miles you can drive when the tank is full, the locations $x_1, \cdots, x_n$ of the gas stations along the highway, where $x_i$ indicates the distance from the $i$-th gas station from $A$. Design a greedy algorithm to compute the minimum number of times you need to fill the gas tank.
A string of “(” and “)” is said to be “balanced”, if it satisfies the recursive definition:

- The empty string “” is balanced.
- If $A$ is balanced then $(A)$ is balanced.
- If $A$ and $B$ are balanced, then $AB$ is balanced.

For example, "(()())()" is balanced.

Problem: Given a string of “(” and “)”, our goal is to remove the minimum number of characters so that the residual string is balanced.

Example: $(())(()())()$
Balanced Strings

A string of “(” and “)” is said to be “balanced”, if it satisfies the recursive definition:

- The empty string “” is balanced.
- If \( A \) is balanced then \((A)\) is balanced.
- If \( A \) and \( B \) are balanced, then \( AB \) is balanced.

For example, ”((())())” is balanced.

Problem: Given a string of “(” and “)”, our goal is to remove the minimum number of characters so that the residual string is a balanced.

- Example: (())(())()