알고리즘_HW08_결과보고서

201802088_류지현

1. 구현 아이디어

주어진 데이터의 구간 값들로 순서를 나열할 때, 비용이 최소가 되기 위해서 여러 가지 경우를 확인해보면 구간들의 값이 오름차순으로 나열되어있을 때, 비용이 최소가 되게 된다.

2. Greedy Choice

이 문제에서의 Greedy Choice는 "***가장 값이 작은 구간부터 나열한다.***" 즉 ***"가장 값이 작은 구간부터 선택한다."***

3. Greedy Choice Property

"가장 값이 작은 구간부터 선택하는 것을 포함하는 최적 해가 반드시 존재한다."

4. Greedy Choice Property Proof

공집합이 아닌 Subproblem S_k을 생각해보자. l_m은 $l_m := \min\{l_i \in S_k\}$인 원소이다. A_k를 S_k의 가장 큰 부분집합이라 하자. 이때, l_j는 A_k의 가장 구간길이가 짧은 원소이다.

ⅰ) $l_j = l_m$ 라면 증명 끝.

ⅱ) $l_j \neq l_m$ 라 하자. 그러면 $l_m < l_j$이므로 $A_k^{'} = A_k - \{a_j\} \cup \{a_m\}$이라 하면 최솟값을 갖는 $A_k^{'}$가 존재하게 된다.

따라서, Greedy Choice하는 것이 항상 안전하다.

5. 구현

Algorithm 1 minimumCost

Input: interval data array C

```
cost ← 0
prev ← 0
len ← Sum of C
for i ← 1 to C.length − 1 do
    if len == C[i] then
        continue
    end if
    cost+ = prev + C[i]
    prev+ = C[i]
    len− = C[i]
end for
    return cost
```

주어진 배열 C가 오름차순으로 정렬되어 있고, C의 첫 번째 원소부터 선택하여 cost에 이전 구간 값과 그다음 구간 값을 더한 값을 더한다. len은 모든 구간의 합으로 반복할 때마다 선택한 구간 값을 빼서 남은 구간 값과 일치하면 통과하게 되는데 결국 마지막 구간의 값을 더하지 않는 것과 같아서 continue 또는 break을 하거나 마지막 구간 직전 원소까지만 cost를 계산하면 된다.

6. 결과