

알고리즘 중급 세미나

04: 고급 다이나믹 프로그래밍

연세대학교 전우제^{kiwiyou}

2023.01.04.r1

배낭 문제

- N 개의 물건에 무게 w_i 와 가치 v_i 가 있을 때, 무게 합이 W 를 넘지 않도록 물건을 몇 개 골라 가치 합을 최대화
- 여기서는 w_i 가 모두 정수인 경우만 고려
- **입력 크기**에 대한 다항 시간에 푸는 알고리즘은 발견되지 않음
- 탐욕적인 접근이 통하기 어려워 보이므로 DP로 접근

배낭 문제

- $f(i, w)$ = i 번째 물건까지 고려했을 때, 정확히 무게 w 를 사용하여 얻을 수 있는 최대 가치
- $f(i + 1, w) = \max(f(i, w), f(i, w - w_{i+1}) + v_i)$
- 시간 $\mathcal{O}(NW)$, 공간 $\mathcal{O}(NW)$
- 배열을 왼쪽부터 덮어씌우면서 공간 $\mathcal{O}(W)$ 에 가능

배낭 문제

- N 가지의 물건에 무게 w_i , 가치 v_i , 개수 c_i 가 있을 때, 무게 합이 W 를 넘지 않도록 물건을 몇 개 골라 가치 합을 최대화
- $C = \sum_{i=1}^N c_i$ 개의 물건이 하나씩 있다고 생각하고 풀기
- $O(CW)$

배낭 문제

- N 가지 물건에 무게 w_i 와 가치 v_i 가 있을 때, 무게 합이 W 를 넘지 않도록 물건을 몇 가지 골라 원하는 개수만큼 선택해 가치 합을 최대화

- $$f(i + 1, w) = \max(\begin{aligned} &f(i, w), \\ &f(i, w - w_{i+1}) + v_i, \\ &f(i + 1, w + w_{i+1}) + v_i \end{aligned})$$

- $\mathcal{O}(NW)$

배낭 문제

- 그 외 최적화 기법은 <https://infossm.github.io/blog/2023/03/18/Knapsack/> 참고

과제

- [17845 수강 과목](#)
- [23257 비트코인은 신이고 나는 무적이다](#)
- [27163 벚꽃 내리는 시대에 결투를](#)

LIS

- 부분 수열^{Subsequence}: 수열 A 에서 몇 개의 원소를 지워 만든 수열
- 가장 긴 증가하는 부분 수열^{Longest Increasing Subsequence}
- 마찬가지로 탐욕법 적용이 어려움

LIS

- $f(i)$: i 번째 원소를 끝으로 하는 가장 긴 증가하는 부분 수열의 길이
- $$f(i) = \max_{\substack{j < i \\ A_j < A_i}} f(j) + 1$$
- 시간 $\mathcal{O}(N^2)$ 공간 $\mathcal{O}(N)$

LIS

- $g(i, L)$: i 번째 원소까지 봤을 때, 길이가 L 인 LIS의 끝 값 중 가장 작은 값
- $f(i) = \max_{g(i-1, L) < A_i} L + 1$
- $g(i, L)$ 은 L 이 길어질수록 증가하므로, 이분 탐색을 이용
- $g(i, L)$ 을 담은 리스트 M 을 만들고, $f(i)$ 가 M 의 길이보다 커질 때마다 M 의 뒤쪽에 i 를 추가
- 그렇지 않으면 $M[f(i)]$ 를 i 로 업데이트 (A_j 가 더 작은 쪽을 남김)

과제

- [12015 가장 긴 증가하는 부분 수열 2](#)
- [12014 주식](#)

해답의 역추적

- DP 문제에서 최적해 혹은 경우의 수를 찾고, **그 예를 하나 찾아야 할 때**
- 이전 상태를 저장
 - DP 테이블의 값을 업데이트할 때마다, 그 값의 이전 상태도 업데이트
- 전이 과정을 거꾸로 따라가기
 - 모든 가능한 상태를 훑으며 전이 후 값이 나오도록 하는 상태를 선택