Dynamic Programming Speedup

14004 구재현

Dynamic Programming

- 알고리즘 디자인 패턴
- 어떠한 문제 상황이 있다. (ex : N개의 동전을 1, 10, 100원 짜리 동전으로 나누고 싶다!)
- 이 문제 상황을 “상태” 라는 수학적인 구조로 표현하며, 상태 간의 관계를 분석한다. (ex : N -> (N-1, N-10, N-100))
- 상태간의 점화식을 작성해서, 문제를 해결한다.
- **정보과학이 자연과 사회를 모델링 하는 중요한 방법 중 하나**

Dynamic Programming 2

- Dynamic Programming은 정보과학의 전부.. 는 아니지만, 아무튼 굉장히 중요한 모델링 기법
- 고로 최적화도 당연히 중요.
- 6가지 최적화 예시를 들고 왔는데 아이디어가 도움이 되길 바람

- 0. Dynamic Programming and Optimal Substructures
- 1. Speedup using Data Structures I : Prefixes
- 2. Speedup using Matrices
- 3. Speedup using Data Structures II : Segment Tree
- 4. Speedup using Monotonicity I : D&C Optimization
- 5. Speedup using Data Structures III : Convex Hull Trick
- 6. Speedup using Monotonicity II : Other Schemes

#1. Prefixes

- dp[i] = dp[i-1] + dp[i-2] … dp[i-k];
- 시간 복잡도 = O(nk)
- Sum[i] = Sum(j <= i) dp[j] 라 정의!
- Sum[i] - Sum[i-1] = Sum[i-1] - Sum[i-k-1]
- Sum[i] = 2 * Sum[i-1] - Sum[i-k-1]
- 고등수학 때 보지 않았나요? 부분합의 힘!
- O(n)에 계산 가능

연습 문제 1

- 최댓값, 최솟값 역시 비슷한 최적화가 가능하지만 설명은 생략한다.

- http://www.koistudy.net/?mid=prob_page&NO=544
- http://www.koistudy.net/?mid=prob_page&NO=144
- http://www.koistudy.net/?mid=prob_page&NO=380

#2. Matrix Multiplication

- dp[i] = dp[i-1] + dp[i-2], 계산 = O(n)
- 너무 느리다 ㅠㅠ
- 행렬식으로 표현!
- 칠판을 참고
- 행렬의 제곱은 O(2 ^ 3).

#2. Matrix Multiplication

- dp[i] = Sum(a[j] * dp[i-j]) (j <= K)
- O(K^3lgN)
- 특수한 경우에는, 행렬 곱셈 안쓰고도 K^2lgN이 가능함
- 특수하지 않더라도, K^2lgN / KlgKlgN에 계산할 수 있는 방법이 존재함. (left as an exercise)

연습 문제 2

- http://koistudy.net/?mid=prob_page&NO=982
- http://koistudy.net/?mid=prob_page&NO=1321

#3. Segment Tree

- 예시 : LIS
- dp[i] = (a[j] < a[i], j < i) Max(dp[j]) + 1
- a의 범위가 충분히 작고 세그먼트 트리가 있다면??
- i를 증가시켜 가면서 처리, a[i] 기준으로 트리에 삽입
- dp[i] = RMQ(0, a[i] - 1) + 1 .. !!!
- Update(a[i], dp[i]);

연습 문제 3

- http://koistudy.net/?mid=prob_page&NO=321
- http://koistudy.net/?mid=prob_page&NO=1213
- https://www.acmicpc.net/problem/2515
- https://www.acmicpc.net/problem/5466
- http://oj.uz/problems/view/JOI14_pinball

#4. Divide & Conquer

- http://koistudy.net/?mid=prob_page&NO=2044
- 첫번째 목공소, 두번째 목공소를 모두 시도하면 n^2개의 가짓수가 있음.
- 첫번째 목공소의 위치 = i라 할때 두번째 목공소의 위치를 정할 수 있다. opt(i) 라고 하자 (i에 대한 함수)
- opt(i) <= opt(i+1) 이라는 관계식이 있는데, 이 성질을 사용해서 nlgn번의 호출만 가지고 문제를 풀 수 있다??

How to??

- divide_conquer([s, e] / [s', e']){ <- 정의역과 치역!
- int m = (s+e) / 2;
- for(int i=s'; i<=e'; i++){
- try all possibilities and get opt(m); } (opt(m)이 여러개면 가장 작은 거)
- divide_conquer([s, m-1], [s', opt(m)]); <<< opt(m-1) <= opt(m)
- divide_conquer([m+1, e], [opt(m), e']); <<< opt(m+1) >= opt(m)
- 치역이 줄어들고 복잡도가 줄어들 거 같은데 실제로 줄어듬
- Total Complexity : O(nlgn)

k개의 목공소라면?

- dp[k][i] = Min(dp[k-1][j] + Cost(j + 1, i));
- dp[k]에 대해서도 opt들이 존재할텐데??
- 고로, K번 NlgN 분할정복을 시도하면 된다
- 시간 복잡도는 KNlgN. 이 문제는 K = 2라 NlgN.
- (좀있다가 이 문제의 KN 풀이를 설명해줌 ㅎ)

연습 문제 4

- http://koistudy.net/?mid=prob_page&NO=2044
- http://www.koistudy.net/?mid=prob_page&NO=1304
- https://www.acmicpc.net/problem/10848

#5. Convex Hull Trick

- 이제는 꽤 유명한가???
- dp[i] = Min(dp[j] + a[i] * b[j]) 라는 점화식이 있다면?
- 선분을 관리하는 자료구조를 만듬

연습 문제 5

- http://koistudy.net/?mid=prob_page&NO=355
- https://www.acmicpc.net/problem/4008

- http://koistudy.net/?mid=prob_page&NO=1065
- https://www.acmicpc.net/problem/5254
- https://www.acmicpc.net/problem/10067
- https://www.acmicpc.net/problem/4220

생각해 볼만한 문제

- 이런 문제는 어떻게 풀까??

- https://www.acmicpc.net/problem/5878

- http://koistudy.net/?mid=prob_page&NO=2049