

CS.20002 문제해결기법

1주차 – 동적 계획법

TA 오태인

2026년 9월 11일

기말고사 안내

- 기말고사는 12/11과 12/18에 진행하며, 총점은 두 시험 점수의 합입니다.
- 두 시험 모두 13:00–16:00에 진행합니다. 조기 퇴실은 가능하지만 늦은 입실은 **절대** 불가능합니다. 12:45까지 입실을 권장합니다.
- 장소는 미정이며, N1 1층에서 진행할 가능성이 높습니다.
- 두 시험에 모두 응시할 필요는 없습니다. 응시하지 않은 시험은 0점으로 계산합니다.
- ICPC 해외 리저널 등의 사유로 한 시험에만 불참하는 경우의 처리 방침은 추후 공지하겠습니다.

- KLMS 공지사항에 'C++ 문제 풀이 팁 모음'을 올렸습니다.
- TA들이 평소 유용하게 사용하는 팁과 과제 제출물을 보며 전하고 싶었던 조언을 모았습니다.
- 비정기적으로 내용을 추가할 예정이니 가끔 확인해 주세요. 추가하고 싶은 내용이 있다면 TA에게 연락해 주세요.
- 기말고사에서 이 자료를 볼 수 있도록 허용할 가능성이 있지만, 아직 확정되지 않았습니다.

언어별 제한 안내

- HW0의 문제에는 사용 언어에 따라 서로 다른 제한이 적용됩니다.
- 이는 의도한 설정이 아니며, 현재는 언어별 제한을 조정할 수 있도록 개선되었습니다.
- HW1부터는 모든 언어에 동일한 제한을 적용합니다. 기말고사도 마찬가지입니다.

- STL의 `stack<int>`를 사용하면 됩니다.
- 1번 쿼리는 `push`, 2번은 `pop`, 3번은 `top`으로 처리합니다.
- 1번 쿼리에서만 x 를 입력받아야 합니다.
- Python의 `st = st[:-1]`처럼 매번 스택 전체를 순회하거나 복사하면 시간복잡도는 $\mathcal{O}(NQ)$ 가 되며, 12점을 받습니다.

- 스택 문제와 비슷하게 STL의 `queue<int>`를 사용하면 됩니다.
- 1번 쿼리는 `push`, 2번은 `pop`, 3번은 `front`로 처리합니다.
- 1번 쿼리에서만 x 를 입력받아야 합니다.
- Python의 `del que[0]`처럼 앞 원소를 지울 때마다 나머지 원소를 이동시키면 시간복잡도는 $O(NQ)$ 가 되며, 12점을 받습니다.

- STL의 `set<int>`를 사용하면 됩니다.
- 1번 쿼리는 `insert`, 2번은 `erase`로 처리합니다.
- 3번 쿼리는 `count(x)`로 x 가 있는지 확인한 뒤, 있으면 `erase`, 없으면 `insert`합니다. 4번 쿼리도 `count(x)`로 처리합니다.
- `find(x) != end()`로 확인할 수도 있지만, `count(x)`가 더 간결하고 `iterator`에 대한 이해가 필요하지 않습니다.
- 3번 쿼리의 삽입·삭제 분기를 잘못 구현하면 68점을 받습니다.

- STL의 `map<int, int>`를 사용하면 됩니다.
- 1번 쿼리는 `mp[i]++`로 값을 늘리고, 2번은 `mp[i]`의 값을 출력하면 됩니다.
- 일반 배열로 구현하면 메모리 제한을 초과하여 9점까지만 받습니다.

- 비교 함수 `cmp`를 작성한 뒤 `sort(arr, arr + n, cmp)`로 정렬하면 됩니다.
- k 로 나눈 몫은 내림차순으로, 몫이 같으면 나머지는 오름차순으로 비교합니다.

```
bool cmp(int a, int b) {  
    if (a / k == b / k) return a % k < b % k;  
    return a / k > b / k;  
}
```

- $\mathcal{O}(N^2)$ 정렬을 직접 구현하면 18점을 받습니다. 몫과 나머지 중 한쪽의 비교만 올바르게 구현하면 각각 8점을 받습니다.

HW0 풀이 – 간단한 그래프 문제

- 지난 수업의 vector 예제와 비슷한 문제입니다.
- 각 정점의 인접 정점을 vector에 저장하고, 쿼리마다 해당 목록을 출력하면 됩니다.
- self-loop (u, u) 가 없고, 서로 다른 간선 $(u_1, v_1), (u_2, v_2)$ 는 $\{u_1, v_1\} \neq \{u_2, v_2\}$ 를 만족합니다. 따라서 중복 제거를 위한 set은 필요하지 않습니다.
- 참고로, 이 조건을 만족하는 그래프를 **단순 그래프**라고 합니다.
- 쿼리의 정점은 모두 다르므로 각 인접 목록은 최대 한 번 출력합니다. 출력하는 정점 수의 합은 다음 차수 합을 넘지 않으므로 $\mathcal{O}(M)$ 입니다.

$$\sum_{u \in V(G)} \deg(u) = 2|E(G)| = 2M.$$

- 정적 인접 행렬 등으로 구현하면 26점을 받습니다.

HW0 풀이 – 하노이 탑

- 시작 봉을 a , 목표 봉을 b , 나머지 봉을 c 라고 합시다.
- 원판 n 개 중 가장 큰 원판을 a 에서 b 로 옮기려면, 나머지 $n - 1$ 개가 모두 c 에 있어야 합니다.
- 따라서 다음 순서로 원판을 옮기면 됩니다.
 - ① $n - 1$ 개의 원판을 a 에서 c 로 옮깁니다.
 - ② 가장 큰 원판 하나를 a 에서 b 로 옮깁니다.
 - ③ $n - 1$ 개의 원판을 c 에서 b 로 옮깁니다.
- $n = 1$ 이면 바로 옮깁니다. $n > 1$ 이면 위 과정을 재귀 함수로 구현합니다.

- 원판 n 개를 옮기는 횟수를 a_n 이라고 하면 다음 점화식을 얻습니다.

$$a_1 = 1, \quad a_n = 2a_{n-1} + 1 \implies a_n = 2^n - 1.$$

- $N \leq 18$ 이면 최대 $2^{18} - 1 = 262\,143$ 번 이동하므로, 10^6 번 이하라는 제한을 만족합니다.
- $N \leq 3$ 인 경우의 이동을 직접 나열하면 6점을 받습니다.
- 가능한 원판 배치는 3^N 개입니다. 각 배치를 상태로 두고, 방문 처리를 하며 BFS로 탐색하면 $N \leq 8$ 까지 해결하여 총 40점을 받습니다.

HW0 풀이 – 이상한 계산기

- 소수 k 에 대한 모듈러 연산을 구현합니다. 저장값 a 는 항상 $0 \leq a < k$ 로 유지합니다.
- C++에서는 음수의 나머지가 음수일 수 있으므로, 뺄셈은 다음과 같이 처리합니다.

`a = (a - x % k + k) % k;`

- 나눗셈은 x 의 역원 $x^{k-2} \bmod k$ 를 구한 뒤 a 에 곱합니다. 지난 수업의 빠른 거듭제곱을 쓰면 역원 하나를 $\mathcal{O}(\log k)$ 에 구할 수 있습니다.
- 나눗셈을 잘못 구현하거나 느리게 구현하면 34점을 받습니다.

HW0 풀이 – 이상한 계산기

- `int x, y;`에 대해 `long long z = x * y;`를 쓰면 곱셈은 `int`로 계산됩니다. overflow가 난 뒤 `z`에 대입해도 복구되지 않습니다.
`1LL * x * y`처럼 곱하기 전에 타입을 바꿔야 합니다.
- `long long x, y, z;`라도 `x * y * z % k`는 overflow가 난 뒤 나머지를 구할 수 있습니다. 각 값이 10^9 이하인 이 문제에서는 곱할 때마다 나머지를 구합니다.
$$x * y \% k * z \% k$$
- 매번 이렇게 쓰는 것이 불편하다면 `mint(modint)` 같은 커스텀 struct를 사용할 수 있습니다. KLMS에 올린 ‘C++ 문제 풀이 팁 모음’을 참고해 주세요.

HW0 풀이 – 간단한 카운팅 문제

- 가장 단순한 풀이는 4중 for문으로 네 정점을 순서대로 고르고, 4-cycle을 이루는지 확인하는 것입니다.
- 각 변수가 모든 정점을 도는 구현의 시간복잡도는 $\mathcal{O}(N^4)$ 입니다. $N = 200$ 에서도 $200^4 = 1.6 \times 10^9$ 번 반복합니다.
- 사이클에서 마주 보는 두 정점 u, v 를 고정합시다. 공통 이웃 x 개 중 두 개를 고르면 4-cycle이 하나 만들어집니다. 그 개수는 다음과 같습니다.

$$\binom{x}{2} = \frac{x(x-1)}{2}$$

- 모든 정점 쌍 $u < v$ 에 대해 이 값을 더합니다. 각 사이클은 마주 보는 정점 쌍이 두 개이므로 두 번씩 세어집니다. 마지막에 합을 2로 나눕니다.

HW0 풀이 – 간단한 카운팅 문제

- u, v 를 고른 뒤 모든 정점을 훑어 공통 이웃을 세면 $\mathcal{O}(N^3)$ 입니다. 이 풀이로 50점을 받을 수 있습니다.
- 모든 정점 대신 u 의 이웃만 보면서 v 에도 인접하는지 확인합니다. 인접 행렬을 함께 사용하면 확인 한 번에 $\mathcal{O}(1)$ 이 걸립니다.
- u 를 고정하고 모든 v 를 처리하는 데 $\mathcal{O}(N \deg(u))$ 가 걸립니다. 차수의 합이 $2M$ 이므로 전체 시간복잡도는 다음과 같습니다.

$$\mathcal{O}\left(N \sum_{u \in V(G)} \deg(u)\right) = \mathcal{O}(2NM) = \mathcal{O}(NM)$$

Break Time

- ICPC Seoul Regional에 참가할 팀은 팀원 중 한 명이 아래 구글폼을 작성하고, 제시된 절차대로 등록을 진행해주세요.

<https://forms.gle/pGrDYZVgmMgPNzq38>

- 공식 마감은 9월 22일 오후 5시입니다. 팀 등록에 시간이 걸리고 승인 여부도 즉시 알 수 없으므로, 가능한 한 빨리 신청해 주세요.
- 문제해결기법 TA는 등록을 담당하지 않습니다. 등록 관련 문의는 아래 연락처로 해 주세요.

<https://kaist.run/ko/contact/>

- 수업 내용에 관한 질문은 편하게 말씀해 주세요.



동적 계획법이란?

- 부분 문제의 답을 저장하고 재사용하여 더 큰 문제의 답을 구하는 풀이법이다.
- 영어로는 dynamic programming, 줄여서 DP라고 한다.
- 이 이름을 붙인 사람은 Richard Bellman이다. 수학 연구에 부정적인 국방부 관계자들의 반감을 피하려고 그럴듯한 이름을 골랐다고 회고했다.
- 쉽게 말해, DP는 점화식을 세우고 계산하는 풀이법이다.

동적 계획법이란?

문제 1

다음 점화식으로 정의되는 수열을 피보나치 수열이라고 한다.

$$f_1 = 1, \quad f_2 = 2, \quad f_i = f_{i-1} + f_{i-2} \quad (i \geq 3).$$

$1 \leq N \leq 500\,000$ 이 주어질 때, f_N 을 998244353으로 나눈 나머지를 구하여라.

- 위 문제는 노골적으로 DP를 사용하라는 문제이다.
- 주어진 점화식을 순서대로 계산하면 $\mathcal{O}(N)$ 에 답을 구할 수 있다.

동적 계획법이란?

문제 2

민재는 계단을 N 칸 오르려고 한다. 한 번에 1칸 또는 2칸 오를 수 있을 때, 계단을 오르는 방법의 수를 구하여라.

- 계단 i 칸을 오르는 방법의 수를 a_i 라고 하자. 첫 이동은 1칸 또는 2칸이다.
- 처음에 1칸 오르면 $i - 1$ 칸이 남고, 2칸 오르면 $i - 2$ 칸이 남는다. 따라서 다음 점화식을 얻는다.

$$a_i = a_{i-1} + a_{i-2} \quad (i \geq 3).$$

- $a_1 = 1, a_2 = 2$ 임은 쉽게 계산할 수 있다.
- 이처럼 DP 문제에서는 주어진 문제로부터 점화식을 얻어내는 과정이 필요하다.

예제 1 – 최장 공통 부분수열

문제 3

길이가 각각 5 000 이하인 두 문자열 A, B 가 주어진다.

문자열 x 에서 문자를 0개 이상 지워 문자열 y 를 만들 수 있다면, y 를 x 의 부분수열(subsequence)이라고 한다.

A, B 의 공통 부분수열 중 가장 긴 것의 길이를 구하여라.

- 단순한 풀이는 A 의 모든 부분수열을 만들고, 각각이 B 의 부분수열인지 확인하는 것이다.
- 시간복잡도는 $\mathcal{O}(2^{|A|}|B|)$ 이므로, 길이가 5 000까지 주어지는 이 문제를 풀기에는 너무 느리다.
- DP를 사용해 보자.

예제 1 – 최장 공통 부분수열

- 문자의 인덱스는 1부터 시작한다. A 의 앞 i 개 문자로 이루어진 문자열을 A_i , B 의 앞 j 개 문자로 이루어진 문자열을 B_j 라고 하자.
- 최장 공통 부분수열을 LCS라고 한다. A_i 와 B_j 의 LCS 길이를 $dp[i][j]$ 라고 하자.
- 마지막 문자를 사용하는 방식에 따라 두 경우로 나눌 수 있다.
 - ① $A[i] = B[j]$ 이고, 두 문자를 서로 대응시켜 사용한다.
 - ② $A[i]$ 또는 $B[j]$ 중 적어도 하나를 사용하지 않는다.
- $A[i] = B[j]$ 인 경우:
$$dp[i][j] = \max(dp[i-1][j-1] + 1, dp[i][j-1], dp[i-1][j]).$$
- $A[i] \neq B[j]$ 인 경우:
$$dp[i][j] = \max(dp[i-1][j], dp[i][j-1]).$$

예제 1 – 최장 공통 부분수열

$A[i] = B[j]$: $dp[i][j] = \max(dp[i-1][j-1] + 1, dp[i][j-1], dp[i-1][j])$

$A[i] \neq B[j]$: $dp[i][j] = \max(dp[i-1][j], dp[i][j-1])$

$A = \text{apple}, \quad B = \text{pale}$

$j \backslash i$		0	1	2	3	4	5
		$\emptyset$	a	p	p	l	e
0	$\emptyset$	0	0	0	0	0	0
1	p	0					
2	a	0					
3	l	0					
4	e	0					

빈 문자열과의 LCS 길이는 0이므로 첫 행과 첫 열을 0으로 채운다.

예제 1 – 최장 공통 부분수열

$A[i] = B[j]$: $dp[i][j] = \max(dp[i-1][j-1] + 1, dp[i][j-1], dp[i-1][j])$

$A[i] \neq B[j]$: $dp[i][j] = \max(dp[i-1][j], dp[i][j-1])$

$A = \text{apple}, \quad B = \text{pale}$

$j \backslash i$	0	1	2	3	4	5
	$\emptyset$	a	p	p	l	e
0 $\emptyset$	0	0	0	0	0	0
1 p	0	0				
2 a	0					
3 l	0					
4 e	0					

$i = 1, j = 1$: $a \neq p \quad dp[1][1] = \max(0, 0) = 0$

예제 1 – 최장 공통 부분수열

$A[i] = B[j]$: $dp[i][j] = \max(dp[i-1][j-1] + 1, dp[i][j-1], dp[i-1][j])$

$A[i] \neq B[j]$: $dp[i][j] = \max(dp[i-1][j], dp[i][j-1])$

$A = \text{apple}, \quad B = \text{pale}$

$j \backslash i$		0	1	2	3	4	5
		$\emptyset$	a	p	p	l	e
0	$\emptyset$	0	0	0	0	0	0
1	p	0	0	1			
2	a	0					
3	l	0					
4	e	0					

$i = 2, j = 1$: p = p $dp[2][1] = \max(0 + 1, 0, 0) = 1$

예제 1 – 최장 공통 부분수열

$A[i] = B[j]$: $dp[i][j] = \max(dp[i-1][j-1] + 1, dp[i][j-1], dp[i-1][j])$

$A[i] \neq B[j]$: $dp[i][j] = \max(dp[i-1][j], dp[i][j-1])$

$A = \text{apple}, \quad B = \text{pale}$

$j \backslash i$		0	1	2	3	4	5
		$\emptyset$	a	p	p	l	e
0	$\emptyset$	0	0	0	0	0	0
1	p	0	0	1	1		
2	a	0					
3	l	0					
4	e	0					

$i = 3, j = 1$: p = p $dp[3][1] = \max(0 + 1, 0, 1) = 1$

예제 1 – 최장 공통 부분수열

$A[i] = B[j]$: $dp[i][j] = \max(dp[i-1][j-1] + 1, dp[i][j-1], dp[i-1][j])$

$A[i] \neq B[j]$: $dp[i][j] = \max(dp[i-1][j], dp[i][j-1])$

$A = \text{apple}, \quad B = \text{pale}$

$j \backslash i$	0	1	2	3	4	5
	$\emptyset$	a	p	p	l	e
0 $\emptyset$	0	0	0	0	0	0
1 p	0	0	1	1	1	
2 a	0					
3 l	0					
4 e	0					

$i = 4, j = 1: l \neq p \quad dp[4][1] = \max(1, 0) = 1$

예제 1 – 최장 공통 부분수열

$A[i] = B[j]$: $dp[i][j] = \max(dp[i-1][j-1] + 1, dp[i][j-1], dp[i-1][j])$

$A[i] \neq B[j]$: $dp[i][j] = \max(dp[i-1][j], dp[i][j-1])$

$A = \text{apple}, \quad B = \text{pale}$

$j \backslash i$		0	1	2	3	4	5
		$\emptyset$	a	p	p	l	e
0	$\emptyset$	0	0	0	0	0	0
1	p	0	0	1	1	1	1
2	a	0					
3	l	0					
4	e	0					

$i = 5, j = 1: e \neq p \quad dp[5][1] = \max(1, 0) = 1$

예제 1 – 최장 공통 부분수열

$A[i] = B[j]$: $dp[i][j] = \max(dp[i-1][j-1] + 1, dp[i][j-1], dp[i-1][j])$

$A[i] \neq B[j]$: $dp[i][j] = \max(dp[i-1][j], dp[i][j-1])$

$A = \text{apple}, \quad B = \text{pale}$

$j \backslash i$		0	1	2	3	4	5
		$\emptyset$	a	p	p	l	e
0	$\emptyset$	0	0	0	0	0	0
1	p	0	0	1	1	1	1
2	a	0	1				
3	l	0					
4	e	0					

$i = 1, j = 2$: a = a $dp[1][2] = \max(0 + 1, 0, 0) = 1$

예제 1 – 최장 공통 부분수열

$A[i] = B[j]$: $dp[i][j] = \max(dp[i-1][j-1] + 1, dp[i][j-1], dp[i-1][j])$

$A[i] \neq B[j]$: $dp[i][j] = \max(dp[i-1][j], dp[i][j-1])$

$A = \text{apple}, \quad B = \text{pale}$

$j \backslash i$		0	1	2	3	4	5
		$\emptyset$	a	p	p	l	e
0	$\emptyset$	0	0	0	0	0	0
1	p	0	0	1	1	1	1
2	a	0	1	1			
3	l	0					
4	e	0					

$i = 2, j = 2$: $p \neq a \quad dp[2][2] = \max(1, 1) = 1$

예제 1 – 최장 공통 부분수열

$A[i] = B[j]$: $dp[i][j] = \max(dp[i-1][j-1] + 1, dp[i][j-1], dp[i-1][j])$

$A[i] \neq B[j]$: $dp[i][j] = \max(dp[i-1][j], dp[i][j-1])$

$A = \text{apple}, \quad B = \text{pale}$

$j \backslash i$		0	1	2	3	4	5
		$\emptyset$	a	p	p	l	e
0	$\emptyset$	0	0	0	0	0	0
1	p	0	0	1	1	1	1
2	a	0	1	1	1		
3	l	0					
4	e	0					

$i = 3, j = 2$: $p \neq a \quad dp[3][2] = \max(1, 1) = 1$

예제 1 – 최장 공통 부분수열

$A[i] = B[j]$: $dp[i][j] = \max(dp[i-1][j-1] + 1, dp[i][j-1], dp[i-1][j])$

$A[i] \neq B[j]$: $dp[i][j] = \max(dp[i-1][j], dp[i][j-1])$

$A = \text{apple}, \quad B = \text{pale}$

$j \backslash i$	0	1	2	3	4	5
	$\emptyset$	a	p	p	l	e
0 $\emptyset$	0	0	0	0	0	0
1 p	0	0	1	1	1	1
2 a	0	1	1	1	1	
3 l	0					
4 e	0					

$i = 4, j = 2$: $l \neq a \quad dp[4][2] = \max(1, 1) = 1$

예제 1 – 최장 공통 부분수열

$A[i] = B[j]$: $dp[i][j] = \max(dp[i-1][j-1] + 1, dp[i][j-1], dp[i-1][j])$

$A[i] \neq B[j]$: $dp[i][j] = \max(dp[i-1][j], dp[i][j-1])$

$A = \text{apple}, \quad B = \text{pale}$

$j \backslash i$	0	1	2	3	4	5
	$\emptyset$	a	p	p	l	e
0 $\emptyset$	0	0	0	0	0	0
1 p	0	0	1	1	1	1
2 a	0	1	1	1	1	1
3 l	0					
4 e	0					

$i = 5, j = 2$: $e \neq a \quad dp[5][2] = \max(1, 1) = 1$

예제 1 – 최장 공통 부분수열

$A[i] = B[j]$: $dp[i][j] = \max(dp[i-1][j-1] + 1, dp[i][j-1], dp[i-1][j])$

$A[i] \neq B[j]$: $dp[i][j] = \max(dp[i-1][j], dp[i][j-1])$

$A = \text{apple}, \quad B = \text{pale}$

$j \backslash i$	0	1	2	3	4	5
	$\emptyset$	a	p	p	l	e
0 $\emptyset$	0	0	0	0	0	0
1 p	0	0	1	1	1	1
2 a	0	1	1	1	1	1
3 l	0	1				
4 e	0					

$i = 1, j = 3: a \neq l \quad dp[1][3] = \max(0, 1) = 1$

예제 1 – 최장 공통 부분수열

$A[i] = B[j]$: $dp[i][j] = \max(dp[i-1][j-1] + 1, dp[i][j-1], dp[i-1][j])$

$A[i] \neq B[j]$: $dp[i][j] = \max(dp[i-1][j], dp[i][j-1])$

$A = \text{apple}, \quad B = \text{pale}$

$j \backslash i$		0	1	2	3	4	5
		$\emptyset$	a	p	p	l	e
0	$\emptyset$	0	0	0	0	0	0
1	p	0	0	1	1	1	1
2	a	0	1	1	1	1	1
3	l	0	1	1			
4	e	0					

$i = 2, j = 3$: $p \neq l \quad dp[2][3] = \max(1, 1) = 1$

예제 1 – 최장 공통 부분수열

$A[i] = B[j]$: $dp[i][j] = \max(dp[i-1][j-1] + 1, dp[i][j-1], dp[i-1][j])$

$A[i] \neq B[j]$: $dp[i][j] = \max(dp[i-1][j], dp[i][j-1])$

$A = \text{apple}, \quad B = \text{pale}$

$j \backslash i$		0	1	2	3	4	5
		$\emptyset$	a	p	p	l	e
0	$\emptyset$	0	0	0	0	0	0
1	p	0	0	1	1	1	1
2	a	0	1	1	1	1	1
3	l	0	1	1	1		
4	e	0					

$i = 3, j = 3$: $p \neq l \quad dp[3][3] = \max(1, 1) = 1$

예제 1 – 최장 공통 부분수열

$A[i] = B[j]$: $dp[i][j] = \max(dp[i-1][j-1] + 1, dp[i][j-1], dp[i-1][j])$

$A[i] \neq B[j]$: $dp[i][j] = \max(dp[i-1][j], dp[i][j-1])$

$A = \text{apple}, \quad B = \text{pale}$

$j \backslash i$		0	1	2	3	4	5
		$\emptyset$	a	p	p	l	e
0	$\emptyset$	0	0	0	0	0	0
1	p	0	0	1	1	1	1
2	a	0	1	1	1	1	1
3	l	0	1	1	1	2	
4	e	0					

$i = 4, j = 3$: $l = l \quad dp[4][3] = \max(1 + 1, 1, 1) = 2$

예제 1 – 최장 공통 부분수열

$A[i] = B[j]$: $dp[i][j] = \max(dp[i-1][j-1] + 1, dp[i][j-1], dp[i-1][j])$

$A[i] \neq B[j]$: $dp[i][j] = \max(dp[i-1][j], dp[i][j-1])$

$A = \text{apple}, \quad B = \text{pale}$

$j \backslash i$	0	1	2	3	4	5
	$\emptyset$	a	p	p	l	e
0 $\emptyset$	0	0	0	0	0	0
1 p	0	0	1	1	1	1
2 a	0	1	1	1	1	1
3 l	0	1	1	1	2	2
4 e	0					

$i = 5, j = 3: e \neq l \quad dp[5][3] = \max(2, 1) = 2$

예제 1 – 최장 공통 부분수열

$A[i] = B[j]$: $dp[i][j] = \max(dp[i-1][j-1] + 1, dp[i][j-1], dp[i-1][j])$

$A[i] \neq B[j]$: $dp[i][j] = \max(dp[i-1][j], dp[i][j-1])$

$A = \text{apple}, \quad B = \text{pale}$

$j \backslash i$		0	1	2	3	4	5
		$\emptyset$	a	p	p	l	e
0	$\emptyset$	0	0	0	0	0	0
1	p	0	0	1	1	1	1
2	a	0	1	1	1	1	1
3	l	0	1	1	1	2	2
4	e	0	1				

$i = 1, j = 4$: $a \neq e \quad dp[1][4] = \max(0, 1) = 1$

예제 1 – 최장 공통 부분수열

$A[i] = B[j]$: $dp[i][j] = \max(dp[i-1][j-1] + 1, dp[i][j-1], dp[i-1][j])$

$A[i] \neq B[j]$: $dp[i][j] = \max(dp[i-1][j], dp[i][j-1])$

$A = \text{apple}, \quad B = \text{pale}$

$j \backslash i$		0	1	2	3	4	5
		$\emptyset$	a	p	p	l	e
0	$\emptyset$	0	0	0	0	0	0
1	p	0	0	1	1	1	1
2	a	0	1	1	1	1	1
3	l	0	1	1	1	2	2
4	e	0	1	1			

$i = 2, j = 4$: $p \neq e \quad dp[2][4] = \max(1, 1) = 1$

예제 1 – 최장 공통 부분수열

$A[i] = B[j]$: $dp[i][j] = \max(dp[i-1][j-1] + 1, dp[i][j-1], dp[i-1][j])$

$A[i] \neq B[j]$: $dp[i][j] = \max(dp[i-1][j], dp[i][j-1])$

$A = \text{apple}, \quad B = \text{pale}$

$j \backslash i$		0	1	2	3	4	5
		$\emptyset$	a	p	p	l	e
0	$\emptyset$	0	0	0	0	0	0
1	p	0	0	1	1	1	1
2	a	0	1	1	1	1	1
3	l	0	1	1	1	2	2
4	e	0	1	1	1		

$i = 3, j = 4$: $p \neq e \quad dp[3][4] = \max(1, 1) = 1$

예제 1 – 최장 공통 부분수열

$A[i] = B[j]$: $dp[i][j] = \max(dp[i-1][j-1] + 1, dp[i][j-1], dp[i-1][j])$

$A[i] \neq B[j]$: $dp[i][j] = \max(dp[i-1][j], dp[i][j-1])$

$A = \text{apple}, \quad B = \text{pale}$

$j \backslash i$		0	1	2	3	4	5
		$\emptyset$	a	p	p	l	e
0	$\emptyset$	0	0	0	0	0	0
1	p	0	0	1	1	1	1
2	a	0	1	1	1	1	1
3	l	0	1	1	1	2	2
4	e	0	1	1	1	2	

$i = 4, j = 4: l \neq e \quad dp[4][4] = \max(1, 2) = 2$

예제 1 – 최장 공통 부분수열

$A[i] = B[j]$: $dp[i][j] = \max(dp[i-1][j-1] + 1, dp[i][j-1], dp[i-1][j])$

$A[i] \neq B[j]$: $dp[i][j] = \max(dp[i-1][j], dp[i][j-1])$

$A = \text{apple}, \quad B = \text{pale}$

$j \backslash i$		0	1	2	3	4	5
		$\emptyset$	a	p	p	l	e
0	$\emptyset$	0	0	0	0	0	0
1	p	0	0	1	1	1	1
2	a	0	1	1	1	1	1
3	l	0	1	1	1	2	2
4	e	0	1	1	1	2	3

$i = 5, j = 4$: $e = e \quad dp[5][4] = \max(2 + 1, 2, 2) = 3$

예제 1 - 최장 공통 부분수열

$A[i] = B[j] : dp[i][j] = \max(dp[i-1][j-1] + 1, dp[i][j-1], dp[i-1][j])$

$A[i] \neq B[j] : dp[i][j] = \max(dp[i-1][j], dp[i][j-1])$

$A = \text{apple}, \quad B = \text{pale}$

$j \backslash i$		0	1	2	3	4	5
		$\emptyset$	a	p	p	l	e
0	$\emptyset$	0	0	0	0	0	0
1	p	0	0	1	1	1	1
2	a	0	1	1	1	1	1
3	l	0	1	1	1	2	2
4	e	0	1	1	1	2	3

$dp[5][4] = 3$ 이므로 답은 3이다. 예: ale, ple.

예제 1 – 최장 공통 부분수열

$A[i] = B[j]$: $dp[i][j] = \max(dp[i-1][j-1] + 1, dp[i][j-1], dp[i-1][j])$

$A[i] \neq B[j]$: $dp[i][j] = \max(dp[i-1][j], dp[i][j-1])$

$A = \text{apple}, \quad B = \text{pale}$

$j \backslash i$		0	1	2	3	4	5
		$\emptyset$	a	p	p	l	e
0	$\emptyset$	0	0	0	0	0	0
1	p	0	0	1	1	1	1
2	a	0	1	1	1	1	1
3	l	0	1	1	1	2	2
4	e	0	1	1	1	2	3

예제 1 – 최장 공통 부분수열

- 점화식에 들어가는 인자는 여러 개일 수 있다. 보통 인자가 n 개인 DP를 n 차원 DP라고 부른다.
- 여러 점화식이 서로 연결된 경우도 있다.

$$\begin{aligned}a_i &= b_{i-1} + a_{i-2}^2, \\b_i &= 2c_{i-2} + a_{i-1}, \\c_i &= 6a_{i-1} + b_{i-1}.\end{aligned}$$

- a_i, b_i, c_i 를 각각 $dp[0][i], dp[1][i], dp[2][i]$ 에 저장하면 위 점화식도 2차원 DP로 구현할 수 있다.
- ‘ n 차원 DP’는 인자의 수에 따라 편의상 붙인 이름이다. 차원 자체에 특별한 의미가 있는 것은 아니다.

예제 2 – 배낭 문제

문제 4

총무게 W 까지 담을 수 있는 가방과 N 개의 물건이 있다. i 번째 물건의 무게는 a_i , 가격은 b_i 이며, 각 물건은 하나씩만 사용할 수 있다.

가방에 담은 물건의 가격 합의 최댓값을 구하여라.

$$N \leq 100, \quad W \leq 5\,000, \quad b_i \leq 10^9.$$

- i 번째 물건까지 고려했을 때의 답을 값 하나로 저장하기에는 애매하다.
- 그렇다고 ' i 번째부터 j 번째 물건까지'처럼 2차원 상태를 잡는다고 해결되는 것은 아니다.
- $\mathcal{O}(NW)$ 에 풀 수 없을지 생각해 보자.

예제 2 – 배낭 문제

- 1 번부터 i 번까지의 물건을 사용해 무게 합을 j 이하로 만들 때, 가격 합의 최댓값을 $dp[i][j]$ 라고 하자.
- i 번째 물건을 쓰는 경우와 쓰지 않는 경우로 나눈다. $j \geq a_i$ 이면 다음 점화식을 얻는다.

$$dp[i][j] = \max(dp[i-1][j], dp[i-1][j-a_i] + b_i).$$

- $j < a_i$ 이면 i 번째 물건을 담을 수 없으므로 $dp[i][j] = dp[i-1][j]$ 이다.
- 이처럼 DP의 인자가 꼭 ‘몇 번째’를 뜻할 필요는 없다.
- TMI: 이 문제를 knapsack problem이라고 한다.

문제 5

$2 \times N$ 표의 각 칸에 수가 적혀 있다. 상하 또는 좌우로 인접한 두 칸을 동시에 고르지 않으면서, 고른 칸에 적힌 수의 합을 최대화하여라.

- 위아래 칸을 번갈아 고르면 가장 많은 칸을 고를 수 있다.
- 이렇게 고르는 두 가지 방법 중 합이 더 큰 쪽을 답으로 생각할 수 있지만, 다음 반례가 있다.

100	1	1
1	1	100

- 번갈아 고르면 최대 합은 102이다. 두 번째 열을 비우고 100 두 개를 고르면 합이 200이 된다.

예제 3

- 각 열에서 위를 고르거나, 아래를 고르거나, 아무것도 고르지 않는 세 가지 선택을 DP의 상태로 삼자.
- i 번째 열까지 고른 수의 합의 최댓값을 마지막 열의 선택에 따라 나누어 저장한다.
 - $dp[0][i]$: i 번째 열에서 아무것도 고르지 않은 경우
 - $dp[1][i]$: i 번째 열에서 위쪽 칸을 고른 경우
 - $dp[2][i]$: i 번째 열에서 아래쪽 칸을 고른 경우
- i 번째 열의 위쪽 수를 a_i , 아래쪽 수를 b_i 라 하면 다음 점화식을 얻는다.
$$\begin{aligned} dp[0][i] &= \max(dp[0][i-1], dp[1][i-1], dp[2][i-1]), \\ dp[1][i] &= \max(dp[0][i-1], dp[2][i-1]) + a_i, \\ dp[2][i] &= \max(dp[0][i-1], dp[1][i-1]) + b_i. \end{aligned}$$
- 답은 $\max(dp[0][N], dp[1][N], dp[2][N])$ 이다.
시간복잡도는 $\mathcal{O}(N)$ 이다.

- 9/18은 카포전, 9/25는 추석 연휴로 휴강합니다.
- 다음 수업은 10/2이며, HW0 제출 기한은 10/1 23:59:00입니다.
- 다음 수업에는 HW0 부정행위 의심자 면담이 예정되어 있습니다.
- 연휴 동안 과제를 즐겁게 고민할 수 있도록 HW1에는 상당히 어려운 문제들도 넣었습니다. 다음 수업은 HW1 풀이만 진행합니다.