

## <알고리즘 과제> 방향그래프

문제: 심층문제 15-4 “에어텔 일반화”

프로그램 요구사항:

- 1) 원문제에서 작성 요구하는 의사코드는 과제에서 제외함.
- 2) **airtelDC**(분할통치 버전)과 **airtelDP**(동적프로그래밍 버전)을 구현하고 실행시간을 비교하라.
- 3) 각 해결버전의 정방향, 역방향 버전 가운데 자유롭게 하나를 선택하여 구현하라.
- 4) 출력에서: 최종적으로 다음과 같이 인쇄되어야 한다  
※ 참고: 1) 아래는 **MAX** = 30인 경우의 인쇄결과임.  
2) X.XXXXXXXXXX는 cputime in milliseconds.

n	s	d	mincost	version	cputime
6	0	4	XXX	DC	X.XXXXXXXXXX
6	0	4	XXX	DP	X.XXXXXXXXXX
6	2	5	XXX	DC	X.XXXXXXXXXX
6	2	5	XXX	DP	X.XXXXXXXXXX
6	2	4	XXX	DC	X.XXXXXXXXXX
6	2	4	XXX	DP	X.XXXXXXXXXX
30	1	28	XXX	DC	X.XXXXXXXXXX
30	1	28	XXX	DP	X.XXXXXXXXXX

주의:

- 1) 반드시 아래 가이드라인의 **main** 함수 내용과 동일하게 실행하고 출력을 구하라.
- 2) **n** = 6로 한 테스트실행 결과가 수작업실행 결과와 일치하는지 확인한 후 **n** = **MAX**로 한 본실행을 시도할 것.
- 3) 경로가 아닌 최소비용만을 요구하는 문제이므로, 만약 최소비용 경로가 두 개 이상이라 하더라도 그 중 아무거나 출력하면 된다.
- 4) 어떤 호출(들)에 대한 계산이 너무 빨라 **cputime** = 0으로 나타나는 것은 허용하지만, **n** = **MAX**로 한 본실행 단계에서, **airtelDC** 호출에 대한 계산이 너무 느리거나 너무 빨라 **airtelDP**의 실행 결과와 비교하기에 적당한 **cputime**을 구하기 어려운 경우에 한하여, 전역변수 **MAX** 값 30을 정확히 절반(즉, 15)으로 줄이거나 두 배(즉, 60)로 늘려 실행하여 제출해도 좋다(15, 30, 60 이외 다른 수 사용은 불가!!).

힌트: 주요 함수 설계

※ 참고: **airtelDC**와 **airtelDP**의 상세 내용은 스스로 작성할 것.

global integer **MAX** = 30 {size of the arrays **A** and **H**}

Alg **main**()

{initialize **A** and **H**}

1. **A**[0], **A**[1]  $\leftarrow$  0, 1

2. for **i**  $\leftarrow$  2 to **MAX** - 1

**A**[**i**]  $\leftarrow$  **A**[**i** - 1] + **A**[**i** - 1]%5 + 3 {**A** = {0, 1, 5, 8, 14, 21, 25, 28, 34, 41, ...}}

3. **H**[0], **H**[1]  $\leftarrow$  0, 5

4. for **i**  $\leftarrow$  2 to **MAX** - 1

**H**[**i**]  $\leftarrow$  (**H**[**i** - 1] + **i**)%9 + 1 {**H** = {0, 5, 8, 3, 8, 5, 3, 2, 2, 3, 5, 8, ...}}

5. **print** "n", "s", "d", "mincost", "version", "cputime" {print header}

{Test run for mini input}

5. **n**  $\leftarrow$  6

6. for (**s**, **d**) in {(0, 4), (2, 5), (2, 4)}

{run three (**s**, **d**) pairs for test}

**airtelDC**(**n**, **s**, **d**)

**printCPUTime**()

{print **n**, **s**, **d**, mincost, version, cputime}

**airtelDP**(**n**, **s**, **d**)

**printCPUTime**()

{print **n**, **s**, **d**, mincost, version, cputime}

{Main run for big input}

7. **n**  $\leftarrow$  **MAX**

8. **s**, **d**  $\leftarrow$  1, **n** - 2

{an (**s**, **d**) pair for test}

9. **airtelDC**(**n**, **s**, **d**)

10. **printCPUTime**()

{print **n**, **s**, **d**, mincost, version, cputime}

11. **airtelDP**(**n**, **s**, **d**)

12. **printCPUTime**()

{print **n**, **s**, **d**, mincost, version, cputime}

Alg **airtelDC**(**n**, **s**, **d**)

{divide and conquer version}

input integer **n**, start city **s**, destination city **d**

output minimum cost of travel from city **s** to city **d**

Alg **airtelDP**(**n**, **s**, **d**)

{dynamic programming version}

input integer **n**, start city **s**, destination city **d**

output minimum cost of travel from city **s** to city **d**