



컴퓨터 알고리즘과 실습



학번 : 2016110056

학과 : 불교학부

이름 : 박승원

날짜 : 2017년 3월 15일

문제 1. 동전 교환 문제 (50)

- 문제 1앞에서 설명한 동전교환문제를 순환 알고리즘으로 구현해 보시오 (부분 점수 없음).
- 유사코드를 작성하시오.
- 프로그램을 작성하고 $M=11$, $c=(1,3,5)$, $d=3$ 의 입력을 넣었을 때와 $M=40$, $c=(25,20,15,10,5,1)$ 의 출력(필요한 동전 갯수출력. 필요한 동전 조합은 출력할 필요 없음)을 제출 하시오. (두경우다 답이 맞았을 경우만 점수 인정)

1.

```
M 40
c <- 25,20,15,10,5,1
cc <- 0,0,...
re <- 0,0,...
minx <- 1000
d <- 5
define F(m) :
  M -= c[m]
  cc[m]++
  if (!M) :
    sum <- 0
    for i -> d : sum += cc[i]
    if sum < minc :
      minc = sum
    for i -> d : re[i] = cc[i]
  for i -> d : re[i] = cc[i]
  M += c[m]
  cc[m]--
define R() :
  sum <- 0
```

```

for i -> d : F(i)
for i -> d : sum += re[i]
print coins, sum
for i -> d : print re[i]

```

2.

```

//2016110056 박승원
#include<iostream>
using namespace std;

struct Coin
{
    int M = 40;
    int c[100] = {25,20,10,5,1,0,};
    int cc[100] = {0,}; // coin count
    int re[100] = {0,}; // result
    int minc = 100; // minimum coin
    int d = 5;

    int select (int m) {
        M -= c[m];
        cc[m]++;
        if (!M) { // 중단 조건
            int sum = 0;
            for(int i=0; i<d; i++) sum += cc[i]; // count coin
            if(sum < minc) {
                minc = sum;
                for(int i=0; i<d; i++) re[i] = cc[i]; // copy to result
            }
        }
        for(int i=0; i<d; i++) if(M >= c[i]) select (i); // 브루트 포스로 선택
        M += c[m];
        cc[m]--;
    }

    int result ()
    {
        int sum = 0;
        for(int i=0; i<d; i++) select (i);
        for(int i=0; i<d; i++) sum += re[i];
        cout << " 필요한 동전 갯수 " << sum << " : ";
        for(int i=0; i<d; i++) cout << re[i] << ' ';
        cout << endl;
    }
}

```

```

    }
};

int main()
{
    Coin A, B;

    A.c[2] = 15; A.c[3] = 10; A.c[4] = 5; A.c[5] = 1;
    A.d = 6;
    A.result ();

    B.M=11;
    B.c[0]=1; B.c[1]=3; B.c[2]=5;
    B.d = 3;
    B.result ();
}

```

```

mf-dist/fonts/typel/public/nanumtypel/nanummjmd4.pfb></usr/share/texlive/texmf-
dist/fonts/typel/public/nanumtypel/nanummjmd5.pfb>
Output written on report.pdf (9 pages, 3264448 bytes).
SyncTeX written on report.synctex.gz.
Transcript written on report.log.
evince report.pdf
zezeon@ubuntuZ:~/Programming/basicProgramming$ make tex
g++ 3.cpp -o 3.x -g -fmax-errors=1 -lm -std=c++14
//2016110056 박승원
#include<iostream>
using namespace std;

struct Coin
{
    int M = 40;
    int c[100] = {25,20,10,5,1,0,};
    int cc[100] = {0,}; //coin count
    int re[100] = {0,}; //result
};

----- 문제 3번 실행을 시작합니다. -----
./3.x
필요한 동전 갯수 2 : 1 0 1 0 0 0
필요한 동전 갯수 3 : 1 0 2
----- 문제 3번 실행을 종료합니다. -----

```

3.

문제 2. 선택정렬 (50 점)

- 수업시간에 비순환적(iterative) 선택정렬을 구현해 보았다. 이를 순환적(recursive) 선택 정렬로 바꾸어 보자.

1. 비순환적 선택 정렬을 구현해 보아라. (20점, 부분 점수 없음)

- 유사코드를작성하시오
- 프로그램을 작성하고 입력 A=[30 20 40 10 5 10 30 15]에 대한 정렬 결과를 출력하여라. 단계별로 A 값이 어떻게 변화되는지 (뭔가 배열에 값이 바뀌거나 각단계마다 A 를 출력하면 됨) 와 최종 결과물을 출력할것.

2. 이를 순환적 선택 정렬로 구현해 보아라. (30점, 부분 점수 없음)

- 유사코드를작성하시오
- 프로그램을 작성하고 입력 A=[30 20 40 10 5 10 30 15]에 대한 정렬 결과를 출력하여라. 단계별로 A 값이 어떻게 변화되는지(뭔가 배열에 값이 바뀌거나 각단계마다 A 를 출력하면 됨) 와 최종 결과물을 출력할 것

1.

```
for i 0 -> end of list :  
    for j i+1 -> end of list :  
        if list [j] == list [i] : swap
```

// 2016110056 박승원

```
#include<iostream>
```

```
using namespace std;
```

```
void swap(int* p, int* q) {
```

```
    int tmp = *p;
```

```
    *p = *q;
```

```
    *q = tmp;
```

```
}
```

```
int main()
```

```
{
```

```
    int ar[] = {30, 20, 40, 10, 5, 10, 30, 15};
```

```
    for(int i=0; i<8; i++) for(int j=i+1; j<8; j++) {
```

```
        if(ar[i] > ar[j]) swap(ar +i, ar + j);
```

```

    for(int i=0; i<8; i++) cout << ar[i] << ' ';
    cout << endl;
}
for(int i=0; i<8; i++) cout << ar[i] << ' ';
}

```

```

zezeon@ubuntuZ:~/Programming/basicProgramming$
zezeon@ubuntuZ:~/Programming/basicProgramming$ rm 4.png
zezeon@ubuntuZ:~/Programming/basicProgramming$ make tex
//2016110056 박승원
#include<iostream>
using namespace std;

```

```

void swap(int* p, int* q) {
    int tmp = *p;
    *p = *q;
    *q = tmp;
}

```

----- 문제 4번 실행을 시작합니다. -----

```

./4.x
20 30 40 10 5 10 30 15
20 30 40 10 5 10 30 15
10 30 40 20 5 10 30 15
5 30 40 20 10 10 30 15
5 30 40 20 10 10 30 15
5 30 40 20 10 10 30 15
5 30 40 20 10 10 30 15
5 30 40 20 10 10 30 15
5 20 40 30 10 10 30 15
5 10 40 30 20 10 30 15
5 10 40 30 20 10 30 15
5 10 40 30 20 10 30 15
5 10 40 30 20 10 30 15
5 10 30 40 20 10 30 15
5 10 20 40 30 10 30 15
5 10 10 40 30 20 30 15
5 10 10 40 30 20 30 15
5 10 10 40 30 20 30 15
5 10 10 30 40 20 30 15
5 10 10 20 40 30 30 15
5 10 10 20 40 30 30 15
5 10 10 15 40 30 30 20
5 10 10 15 30 40 30 20
5 10 10 15 30 40 30 20
5 10 10 15 20 40 30 30
5 10 10 15 20 30 40 30
5 10 10 15 20 30 40 30
5 10 10 15 20 30 30 40

```

----- 문제 4번 실행을 종료합니다. -----

```

define F( start_pointer , end_pointer) :
    if start == end : return
    for p from start to end : if(* start > *p) swap(p, start )
    F( start + 1, end)

```

```

//2016110056 박승원
#include<string>
#include<random>
#include<iostream>
using namespace std;

void swap(int* p, int* q) {
    int tmp = *p;
    *p = *q;
    *q = tmp;
}

void selection_sort (int* p, int* q) {
    if(p == q) return;
    for(int* i = p; i != q; i++) if(*p > *i) swap(p, i);
    selection_sort (p+1, q);
}

int main()
{
    int ar[] = {30, 20, 40, 10, 5, 10, 30, 15};
    selection_sort (ar, ar+8);
    for(int i=0; i<8; i++) cout << ar[i] << ' ';
}

```

```

----- 문제 1번 실행을 종료합니다 . -----
//2016110056 박승원
#include<string>
#include<random>
#include<iostream>
using namespace std;

void swap(int* p, int* q) {
    int tmp = *p;
    *p = *q;
    *q = tmp;
}

----- 문제 5번 실행을 시작합니다 . -----
./5.x
5 10 10 15 20 30 30 40 ----- 문제 5번 실행을 종료합니다 . -----
-----

```