[ 게임 알고리즘 ]

**게임 구현을 위해 사용한 자료구조 및 알고리즘**

인하대학교 미래인재개발원 문화콘텐츠프로그래밍

김예슬

**게임 구현을 위해 사용한 자료구조**

- 큐, 스택

#include<stdio.h>

#include<stdlib.h>

#include"p132\_intStack.h"

int Initialize(INTSTACK\_S \*s, int max)

{

s->ptr = 0;

if ((s->stk = (int \*) calloc(max, sizeof(int))) == NULL)

{

s->max = 0;

return -1;

}

s->max = max;

return 0;

}

int Push(INTSTACK\_S \*s, int x)

{

if (s->ptr >= s->max)

return -1; // 스택이 가득 참

s->stk[s->ptr++] = x;

return 0;

}

int Pop(INTSTACK\_S \*s, int \*x)

{

if (s->ptr <= 0)

return -1; // 스택이 비어있음

\*x = s->stk[s->ptr - 1];

s->ptr--;

return 0;

}

int Peek(const INTSTACK\_S \*s, int \*x)

{

if (s->ptr <= 0)

return -1; // 스택이 비어있음

\*x = s->stk[s->ptr - 1];

return 0;

}

void Clear(INTSTACK\_S \*s)

{

s->ptr = 0;

}

int Capacity(const INTSTACK\_S \*s)

{

return s->max;

}

int Size(const INTSTACK\_S \*s)

{

return s->ptr;

}

int IsEmtpy(const INTSTACK\_S \*s)

{

return s->ptr <= 0;

}

int IsFull(const INTSTACK\_S \*s)

{

return s->ptr >= s->max;

}

int Search(const INTSTACK\_S \*s, int x)

{

for (int i = s->ptr - 1; i >= 0; i--)

{

if (s->stk[i] == x)

return i;

}

return -1;

}

void Print(const INTSTACK\_S \*s)

{

for (int i = 0; i < s->ptr; i++)

{

printf("%d", s->stk[i]);

}

putchar('\n');

}

void Terminate(INTSTACK\_S \*s)

{

if (s->stk != NULL)

free(s->stk);

s->max = s->ptr = 0;

}

- 정렬

#include<iostream>

#include<ctime>

#include<cstdlib>

#include"Stopwatch.h"

using namespace std;

void PrintArr(int insertArr[]);

const int arrSize = 100000;

int main()

{

srand((unsigned)time(NULL));

STOPWATCH\_C stopWatch;

int insertArr[arrSize] = { 0 };

for (int i = 0; i < arrSize; i++)

{

insertArr[i] = rand() % arrSize + 1;

}

int temp;

stopWatch.StartTime();

for (int i = 0; i < arrSize - 1; i++)

{

for (int j = i + 1; j > 0; j--)

{

if (insertArr[j - 1] > insertArr[j])

{

temp = insertArr[j - 1];

insertArr[j - 1] = insertArr[j];

insertArr[j] = temp;

// Swap()으로 둘 경우 시간이 너무 오래 걸림

}

else

break;

}

}

stopWatch.EndTime();

cout << stopWatch.GetElapsedTime() << endl;

return 0;

}

void PrintArr(int insertArr[])

{

for (int i = 0; i < arrSize; i++)

{

cout << insertArr[i] << " ";

}

cout << endl;

}