

Data Structure

Week 9
KyuDong SIM

1. 이번 주 실습 내용

- Equivalence Class

Equivalence Class

- 주어진 집합 X 에 대해 X 내부의 동치 관계 \sim 로 주어지는 X 의 부분집합
- $[a] = \{x \in X \mid a \sim x\}$
- 동치 관계의 원소는 다음 3가지를 만족하는 이항 관계이다.
 - 반사 관계 : $a \sim a$
 - 대칭 관계 : $a \sim b \Rightarrow b \sim a$
 - 추이 관계 : $a \sim b, b \sim c \Rightarrow a \sim c$

Find Equivalence Class

- Algorithm
 - 둘의 관계 (i, j) 를 입력 받고 저장
 - 0에서 시작하여 모든 $(0, x)$ 관계를 구함
 - 추이 관계($a \sim b, b \sim c \Rightarrow a \sim c$)를 사용하여 $(0, x)$ 과 (x, y) 로부터 0과 동치류를 찾음
 - ... 1에서 시작하여 위의 과정 반복

Insert and Save

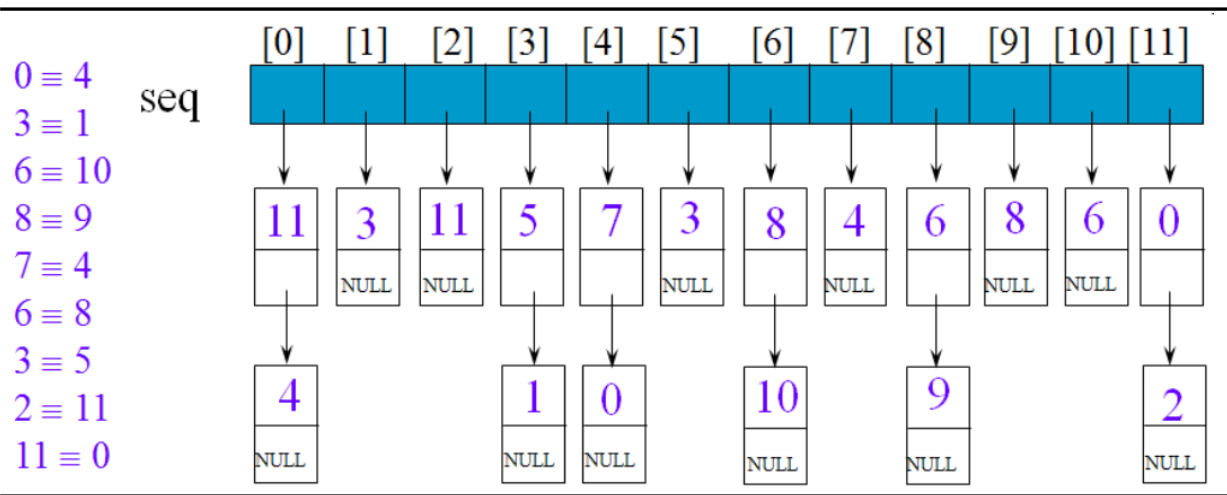


Figure 4.16 : Lists after pairs have been input

Find Equivalence Class

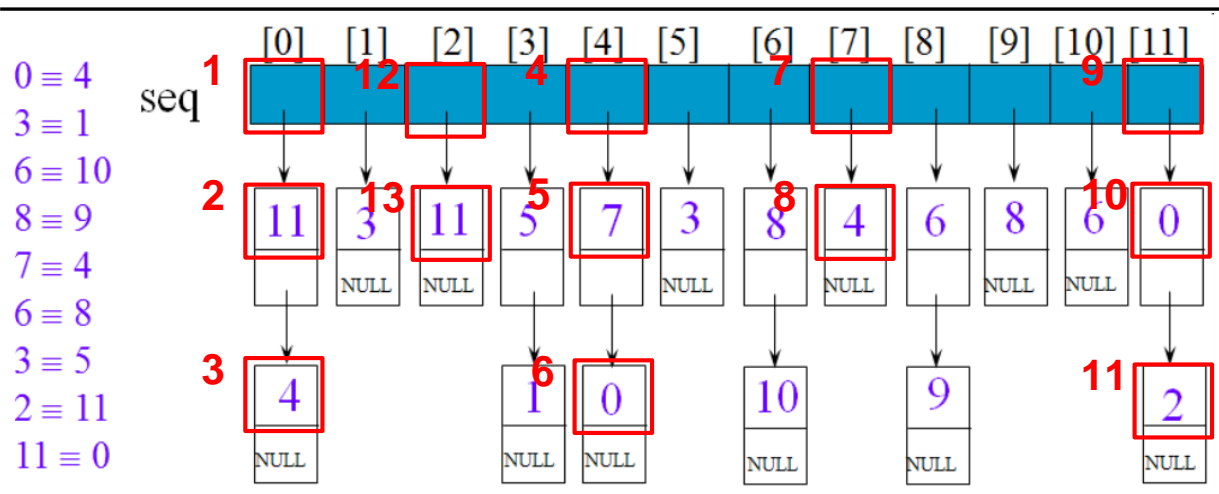


Figure 4.16 : Lists after pairs have been input

결과 예

```

C:\Windows\system32\cmd.exe
Enter the size ( <= 24 ) : 12
Enter a pair of numbers (-1 -1 to quit) : 0 4
Enter a pair of numbers (-1 -1 to quit) : 3 1
Enter a pair of numbers (-1 -1 to quit) : 6 10
Enter a pair of numbers (-1 -1 to quit) : 8 9
Enter a pair of numbers (-1 -1 to quit) : 7 4
Enter a pair of numbers (-1 -1 to quit) : 6 8
Enter a pair of numbers (-1 -1 to quit) : 3 5
Enter a pair of numbers (-1 -1 to quit) : 2 11
Enter a pair of numbers (-1 -1 to quit) : 11 0
Enter a pair of numbers (-1 -1 to quit) : -1 -1

New class:      0    11    4    7    2
New class:      1     3     5
New class:      6     8    10     9
계속하려면 아무 키나 누르십시오 . . .
  
```

Class 개수 입력
 데이터 쌍 입력
 Equivalence Class 출력

Data type

```
#include <stdio.h>
#include <stdlib.h>

#define MAX_SIZE 24
#define FALSE 0
#define TRUE 1

typedef struct node
{
    int data;
    struct node* link;
};

typedef struct node *nodePointer;
```


Input and Save Function

```
int main(void)
{
    short int out[MAX_SIZE];
    nodePointer seq[MAX_SIZE];
    nodePointer x,y,top;
    int i,j,n;

    printf("Enter the size ( <= %d ) : ",MAX_SIZE);
    scanf("%d",&n);
    for(i=0; i<n; i++)
    {
        out[i] = TRUE;
        seq[i] = NULL;
    }

    printf("Enter a pair of numbers (-1 -1 to quit) : ");
    scanf("%d%d",&i,&j);
}
```

Seq[]: 헤더 노드

Out[]: 스캔 여부 확인 변수

Input and Save Function

```
while(i>=0)
{
    x = (struct node *) malloc(sizeof(struct node));
    x->data = j;
    x->link = seq[i];
    seq[i] = x;
    x = (struct node *) malloc(sizeof(struct node));
    x->data = i;
    x->link = seq[j];
    seq[j] = x;
    printf("Enter a pair of numbers (-1 -1 to quit) : ");
    scanf("%d%d",&i,&j);
}
```

Linked list로 구현된 stack과 같이
입력 받은 i, j를 삽입

Print Function

```

/* Phase 2: output the equivalence classes */
for (i = 0; i < n; i++)
    if (out[i]) {
        printf("\nNew class: %5d",i);
        out[i] = FALSE; /* set class to false */
        x = seq[i]; top = NULL; /* initialize stack */
        for(;;) { /* find rest of class */
            while (x) { /* process list */
                j = x->data;
                if (out[j]) {
                    printf("%5d",j); out[j] = FALSE;
                    y = x->link; x->link = top; top = x; x = y;
                }
                else x = x->link;
            }
            if (!top) break;
            x = seq[top->data]; top = top->link;
            /* unstack */
        }
    }
}

```

Push

Pop

생성한 노드를 그대로 Stack으로
사용

Out[]을 이용해 스캔여부를
확인하며 모든 노드를 탐색

Program 4.22_3 : Program to find equivalence classes

제출 및 알림

수업 중 확인 or 메일제출 (학번 써주세요)

메일 제출 :

주소 : (89kdsim@naver.com)

기한 : ~2015-05-04