자료 구조
숙제 #05
트리

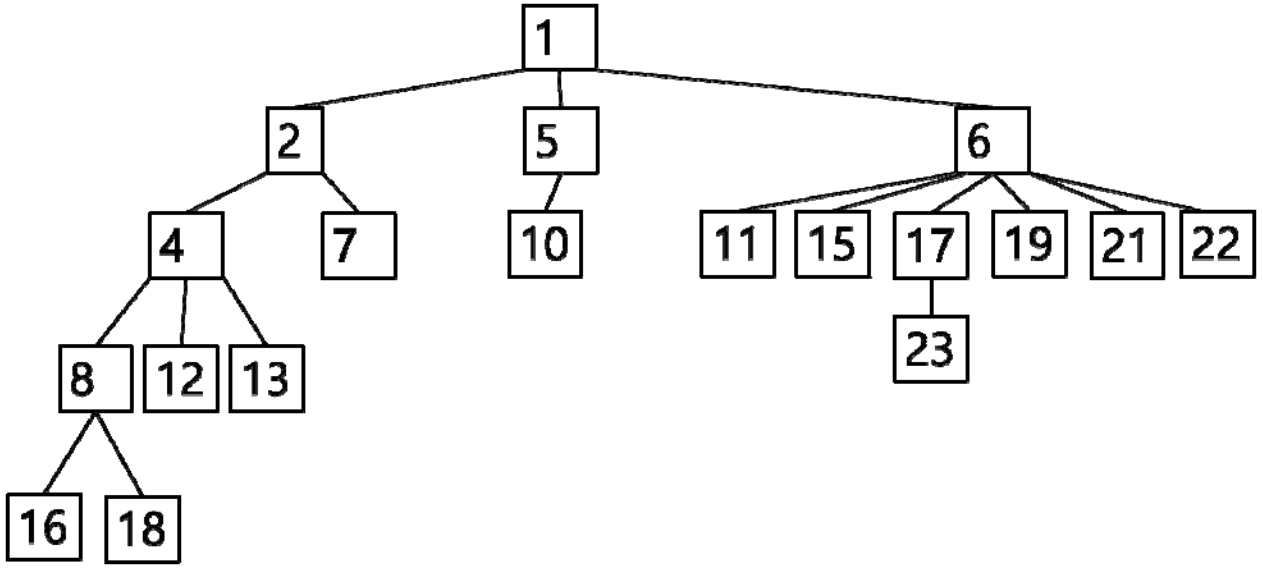
2분반 마감 시간: 5월 3일 오후 11시 59분
1분반 마감 시간: 5월 6일 오후 11시 59분

2018/04/27

컴퓨터과학과
민경하

문제

- 다음과 같은 트리가 다음과 같은 형식으로 test.txt 파일에 저장되어 있다.



test - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
1 2 5 6
2 4 7
5 10
6 11 15 17 19 21 22
4 8 12 13
8 16 18
17 23
```

문제

- 트리에 대한 자료 구조가 다음과 같이 정의되어 있다.

```
typedef class node *nptr;

class node {
    int data;
    int nchilds;
    nptr *childs;
};
```

문제

- 위의 test.txt 파일로부터 한줄씩 트리에 대한 정보를 읽어들여서 트리에 추가하는 과정을 통해서 트리를 완성한다.

```
int main()
{
	int i;
	int n_tok;
	FILE *fp = fopen("test.txt", "r+t");
	char str[256];
	int tok[10];

	node root;
```

문제

- 위의 test.txt 파일로부터 한줄씩 트리에 대한 정보를 읽어들여서 트리에 추가하는 과정을 통해서 트리를 완성한다.

```
	while (fgets(str, 256, fp) != NULL) {
		for (i = 0; i < 10; i++)
			tok[i] = -1;
		sscanf(str, "%d %d %d %d %d %d %d %d %d %d",
			&tok[0], &tok[1], &tok[2], &tok[3], &tok[4],
			&tok[5], &tok[6], &tok[7], &tok[8], &tok[9]);
		for (i = 0, n_tok = 0; i < 10; i++) {
			if (tok[i] >= 0)
				n_tok++;
		}
		root.build(n_tok, tok);
	}

	root.print();
```

문제

- build ( )과정은 3단계로 구분됨.
  (1) Degenerate case → root node에 삽입

```
void node::build(int n, int *cdata)
{
        int i;
        if (this->data < 0) {
                this->data = cdata[0];
                this->nchilds = n - 1;
                this->childs = (nptr *)calloc(this->nchilds, sizeof(nptr));
                for (i = 0; i < this->nchilds; i++) {
                        this->childs[i] = (nptr)malloc(sizeof(node));
                        this->childs[i]->data = cdata[i + 1];
                        this->childs[i]->nchilds = 0;
                        this->childs[i]->childs = NULL;
                }
                return;
        }
```

문제

- build ( )과정은 3단계로 구분됨.

  (2) 삽입할 노드를 찾은 경우

```
        if (this->data == cdata[0]) {
                this->nchilds = n - 1;
                this->childs = (nptr *)calloc(this->nchilds, sizeof(nptr));
                for (int i = 0; i < this->nchilds; i++) {
                        this->childs[i] = (nptr)malloc(sizeof(node));
                        this->childs[i]->data = cdata[i + 1];
                        this->childs[i]->nchilds = 0;
                        this->childs[i]->childs = NULL;
                }
                return;
        }
```

문제

- build ( )과정은 3단계로 구분됨.

  (3) 삽입할 노드가 아니면 그 자식 노드에 대해서 탐색을 수행

```
        for (int i = 0; i < this->nchilds; i++) {
                this->childs[i]->build (n, cdata);
        }
}
```

문제

- print ( )과정은 재귀 호출로 구현.

```
void node::print()
{
        int i;
        printf("[%d] ", this->data);
        for (i = 0; i < nchilds; i++) {
                printf("%d  ", this->childs[i]->data);
        }
        printf("\n");
        for (i = 0; i < nchilds; i++)
                this->childs[i]->print();
}
```

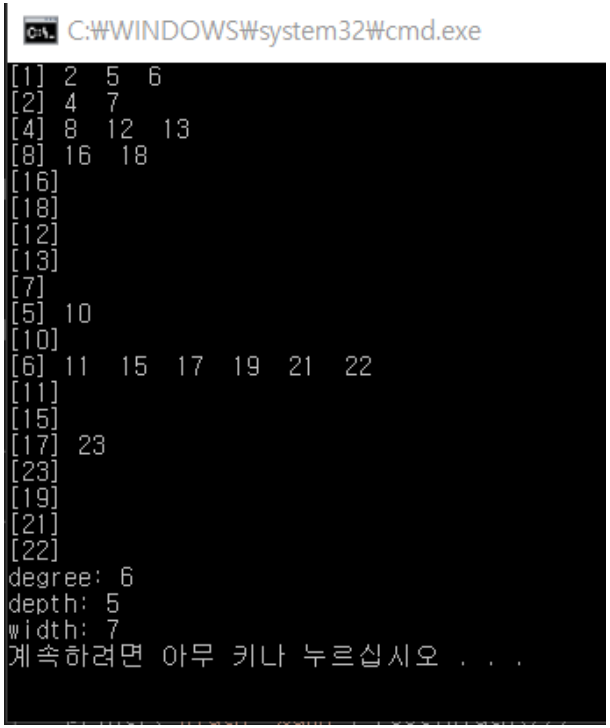
문제

- 위의 트리에 대해서 다음의 3 가지 연산을 구현할 것.
  - degree ( ): 트리의 degree를 출력하시오.
  - depth ( ): 트리의 depth를 출력하시오.
  - width ( ): 트리의 width를 출력하시오.

```
printf("degree: %d\n", root.degree());
printf("depth: %d\n", root.depth());
printf("width: %d\n", root.width());
```

문제

- 위의 트리에 대한 출력:



평가

- degree ( )는 30점
- depth ( )는 30점
- width ( )는 40점

벌점

- 1시간 늦으면 10% 감점
- Copy 또는 copied는
  - 1회 적발은 해당 숙제 0점 처리
  - 2회 적발은 이 과목 F 처리
- 본인이 직접 숙제 하지 않은 경우는 copy 또는 copied와 동일하게 처벌함
  - 반드시 코드를 설명할 수 있을 것