

Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Інститут прикладного системного аналізу
Кафедра системного проектування

ЗВІТ
про виконання комп'ютерного практикуму № 4
з дисципліни «Алгоритми і структури даних»

Виконав: студент 1 курсу групи ДА-83

Цибін Максим Дмитрович

Варіант 27

Прийняв:

Київ – 2018

Зміст

Практична робота 4

1 Завдання

2 Блок схема реалізованої функції

3 Лістинг програми

4 Висновки

ПРАКТИЧНА РОБОТА 4

1 Завдання

Для однозв'язного списку груп (поля: title, amount_of_students) реалізувати функції:

А) Додавання елемента у хвіст списку;

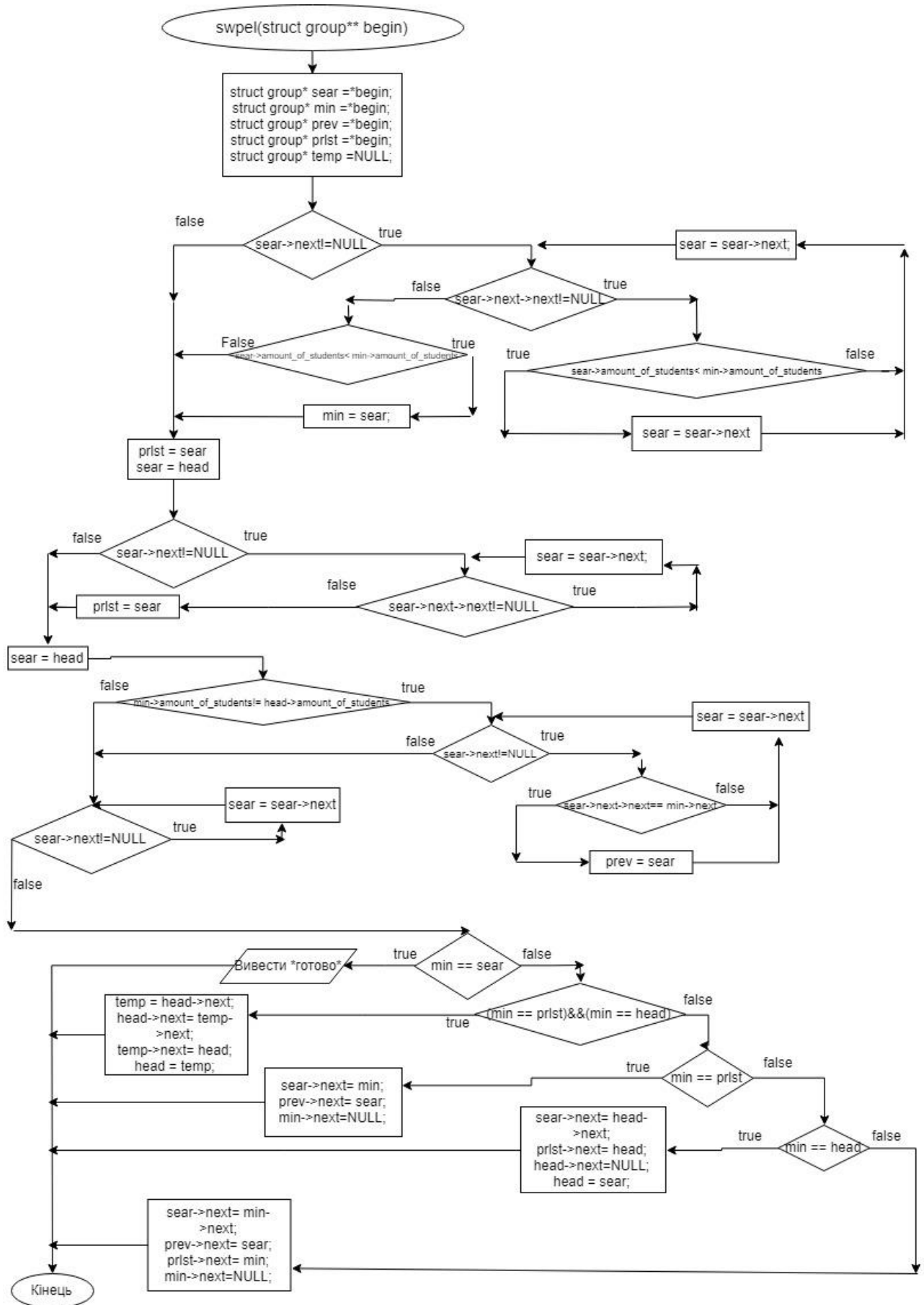
Б) Видалення елемента з голови списку;

В) Поміняти місцями елемент з найменшою кількістю студентів та елемент у хвості;

Г) Надрукувати весь список, формат “title: amount_of_students”;

Д) Видалити весь список.

2 Блок схема реалізованої функції



3 Лістинг програми

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

struct group
{
    char title[6];
    int amount_of_students;
    struct group* next;
};
struct group* head = NULL;

//Добавить элемент в хвост
void addel(struct group **begin, char* tit, int amt)
{
    struct group* tmp = (struct group*) malloc(sizeof(struct group));

    int count = 0;
    while(*tit != '\0')
    {
        tmp->title[count]=*tit;
        tit++;
        count++;
    }
    tmp->amount_of_students = amt;
    tmp->next = NULL;

    struct group *p = head;
    if (p == NULL)
    {
        *begin = tmp;
    }
    else
    {
        while (p->next != NULL)
        {
            p = p->next;
        }
        p->next = tmp;
    }
}
```

```

//Удалить элемент из головы
void delel(struct group **begin)
{
    struct group *del = *begin;
    if (del != NULL)
    {
        head = del->next;
        free(del);
    }
}

//Обмен
void swpel(struct group **begin)
{
    struct group* sear = *begin;
    struct group* min = *begin;
    struct group* prev = *begin;
    struct group* prlst = *begin;
    struct group* temp = NULL;

    //находим минимальный
    if(sear->next != NULL)
    {
        while(sear->next != NULL)
        {
            if(sear->amount_of_students < min->amount_of_students)
                min = sear;
            sear = sear->next;
        }
        if(sear->amount_of_students < min->amount_of_students)
            min = sear;
    }

    //элемент перед последним
    sear = head;
    if(sear->next != NULL)
    {
        while(sear->next->next != NULL)
        {
            sear = sear->next;
        }
        prlst = sear;
    }
}

```

```

//элемент перед минимальным
sear = head;
if(min->amount_of_students != head->amount_of_students)
{
    while(sear->next != NULL)
    {
        if(sear->next->next == min->next)
            prev = sear;
        sear = sear->next;
    }
}
while(sear->next != NULL)
{
    sear = sear->next;
}

//Меняем
if(min == sear) //Если мин последний
{
    printf("\n\n....Already....\n\n");
}else if((min == prlst)&&(min == head)) //Если мин предпоследний и первый
{
    temp = head->next;
    head->next = temp->next;
    temp->next = head;
    head = temp;
}else if(min == prlst) //Если мин предпоследний
{
    sear->next = min;
    prev->next = sear;
    min->next = NULL;
}else if(min == head) //Если мин первый
{
    sear->next = head->next;
    prlst->next = head;
    head->next = NULL;
    head = sear;
}else
{
    sear->next = min->next;
    prev->next = sear;
    prlst->next = min;
    min->next = NULL;
}
}

```

```

//Вывести
void printlist()
{
    struct group *p = head;
    printf("....Group list...\n");
    while(p != NULL)
    {
        printf("\n");
        for(int i = 0; i < 5; i++)
        {
            printf("%c", p->title[i]);
        }
        printf(": %d\n", p->amount_of_students);
        p = p->next;
    }
    printf("\n");
}

//Удалить список
void dellist()
{
    struct group *m = head;
    while(m != NULL)
    {
        delel(&head);
        m = m->next;
    }
    free(m);
}

// Полетели
int main()
{
    int ex = -1;
    while(ex != 0)
    {
        printf(".....MENU.....\n");
        printf("|          Group list          |\n");
        printf("|_____|\n");
        printf("| Press 1 to add element to the tail |\n");
        printf("| Press 2 to delete element from the head |\n");
        printf("| Press 3 to swap the smallest with last |\n");
    }
}

```



```

printf("| Press 4 to print the list          |\n");
printf("| Press 5 to delete the list         |\n");
printf("| Press 6 to exit...                   |\n");
printf(".....\n");
scanf("%d",&ex);
if(ex == 1)
{
    system("cls");
    char tit[6];
    int amt;
    printf("Enter the name of group: \n");
    scanf("%s",tit);
    printf("Enter amount of students: \n");
    scanf("%d",&amt);
    addel(&head,tit,amt);
}
if(ex == 2)
{
    system("cls");
    delel(&head);
}
if(ex == 3)
{
    system("cls");
    swpel(&head);
}
if(ex == 4)
{
    system("cls");
    printlist();
}
if(ex == 5)
{
    system("cls");
    dellist();
}
if(ex == 6)
{
    ex = 0;
}
}

return 0;
}

```

4 Висновок

В ході виконання практичної роботи мною був набутий досвід розробки алгоритмів роз'язку задач. Було отримано навички роботи з динамічними лінійними типами даних – зв'язними списками.