目录

[优先队列重载 1](#_Toc21405)

[set重载 1](#_Toc17434)

[动态开数组 1](#_Toc11905)

图论初始化！！

### 优先队列重载

priority\_queue<int, vector<int>, cmp>s;  
  
struct cmp{  
 bool operator()(const int &a,const int &b){  
 return a>b;   
 }   
};

### set重载

#include <bits/stdc++.h>  
using namespace std;  
  
#define l first  
#define r second   
  
struct cmp{  
 bool operator() (const pair <int, int> &a, const pair<int, int> &b) const{  
 int lena = a.r - a.l + 1;  
 int lenb = b.r - b.l + 1;  
 if(lena == lenb) return a.l < b.l;  
 return lena > lenb;  
 }  
};  
  
int main(){  
 ios :: sync\_with\_stdio(0); cin.tie(0); cout.tie(0);  
 int T;  
 cin >> T;  
 while(T -- ){  
 int n;  
 cin >> n;  
 set<pair<int, int>, cmp> segs;  
 segs.insert({0, n - 1});  
 vector<int> a(n);  
 for(int i = 1; i <= n; ++ i){  
 pair<int, int> cur = \*segs.begin();  
 segs.erase(segs.begin());  
 int id = (cur.l + cur.r) / 2;  
 a[id] = i;  
 if(cur.l < id) segs.insert({cur.l, id - 1});  
 if(id < cur.r) segs.insert({id + 1, cur.r});  
 }  
   
 for(auto it : a) cout << it << " ";  
 cout << endl;  
 }  
}

### 动态开数组

int a[15], n, m;  
cin >> n >> m;  
int (\*b)[m] = (int (\*)[m])a;

**new / delete**

#define M 10U  
  
#define N 20  
  
第一种，可以直接[][]访问。但是内存不连续，不是很推荐使用，除非M \ N都不确定  
  
//定义的时候  
  
int\*\* pNum;//以int为例  
  
pNum = new int\*[M];  
  
for(int i = 0;i < M;i ++){  
 pNum[i]=new int[N];  
}  
  
  
//删除的时候是  
  
for(int j = 0;j < M;j ++){  
 delete []pNum[i];  
}  
  
delete []pNum;

**malloc / free**

#include<stdio.h>   
#include<stdlib.h>   
   
int main() {   
 int \*\*a; //用二级指针动态申请二维数组   
 int i,j;   
 int m,n;   
 printf("请输入行数\n");   
 scanf("%d",&m);   
 printf("请输入列数\n");   
 scanf("%d",&n);   
 a=(int\*\*)malloc(sizeof(int\*)\*m);   
 for(i=0;i<m;i++)   
 a[i]=(int\*)malloc(sizeof(int)\*n);   
 for(i=0;i<m;i++) {  
 for(j=0;j<n;j++) {  
 printf("%p\n",&a[i][j]); //输出每个元素地址，每行的列与列之间的地址时连续的，行与行之间的地址不连续  
 }  
 }  
 for(i=0;i<m;i++)   
 free(a[i]);  
   
 free(a);   
 return 0;   
}

#include<stdio.h>   
#include<stdlib.h>   
   
int main()   
{   
 int i,j;   
 //申请一个3行2列的整型数组   
 int (\*a)[2]=(int(\*)[2])malloc(sizeof(int)\*3\*2);   
 for(i=0;i<3;i++) {  
 for(j=0;j<2;j++) {   
 printf("%p\n",&a[i][j]); //输出数组每个元素地址，每个元素的地址是连续的  
 }  
 }  
   
 free(a);  
 return 0;   
}

**vector**

//二维vector  
vector<vector <int> > ivec(m ,vector<int>(n)); //m\*n的二维vector  
   
//动态创建m\*n的二维vector  
//方法一：  
vector<vector <int> > ivec;  
ivec.resize(m);  
for(int i=0;i<m;i++)   
 ivec[i].resize(n);  
   
//方法二：  
vector<vector <int> > ivec;  
ivec.resize(m,vector<int>(n));