**贪心算法实验1**

**1、部分背包问题**

**问题描述：**

假设山洞中有n种宝物，每种宝物有一定重量w和相应的价值v，毛驴运载能力有限，只能运走m重量的宝物，一种宝物只能拿一样，宝物可分。那么怎么才能使毛驴运走宝物的价值最大呢？

**输入示例：（第一行输入毛驴最大载重量C，宝物重量和价值输入格式自定）**

**30**



**输出示例：（第一行为带走的宝物总价值，第二行为带走的每个宝物的情况）**

总价值为：？

**（2，10，6，3，5，8）** ，其中8带走的是一部分

|  |
| --- |
| Task1.cpp |
| #include <bits/stdc++.h>  #define endl '\n'  using namespace std;  typedef long long ll;  struct Node  {  ll idx;  ll c;  ll v;  };  bool cmp(Node a, Node b) {  if (a.v \* 1.0 / a.c > b.v \* 1.0 / b.c)  return true;  return false;  }  void solve() {  ll mc, n;  cin >> mc >> n;  vector<Node> arr(n);  for (int i = 0; i < n; i++) {  cin >> arr[i].c >> arr[i].v;  arr[i].idx = i + 1;  }  sort(arr.begin(), arr.end(), cmp);  vector<ll> ans;  ll i = 0;  double mv = 0;  bool tag = false;  while (mc > 0 && i < n) {  if (arr[i].c <= mc) {  ans.push\_back(arr[i].idx);  mc -= arr[i].c;  mv += arr[i].v;  } else {  ans.push\_back(arr[i].idx);  tag = true;  mv += arr[i].v \* 1.0 / arr[i].c \* mc;  mc = 0;  }  i++;  }  cout << "MAX Value:" << mv << "\n";  cout << ans[0];  for (ll i = 1; i < ans.size(); i++)  cout << " " << ans[i];  if (tag)  cout << "\*\n";  }  int main() {  int T;  // cin >> T;  T = 1;  while (T--) {  solve();  }  return 0;  } |

**2、乘船问题**

**问题描述：**

某班级春游，组织水上游船活动。现有n个同学，体重为w1、w2、…、wn，游船的最大载重量为C，且最多容许乘坐2人。如果你是组织者，为了节约费用，请计算最少的游船数量和乘船方案。

**输入示例：（第一行为班级人数和游船载重量，第二行为每个人的体重）**

4 85

45 65 40 58

**输出示例：（第一行为最少游船数m，下面m行每行为乘船人的体重）**

3

1: 40 45

2: 58

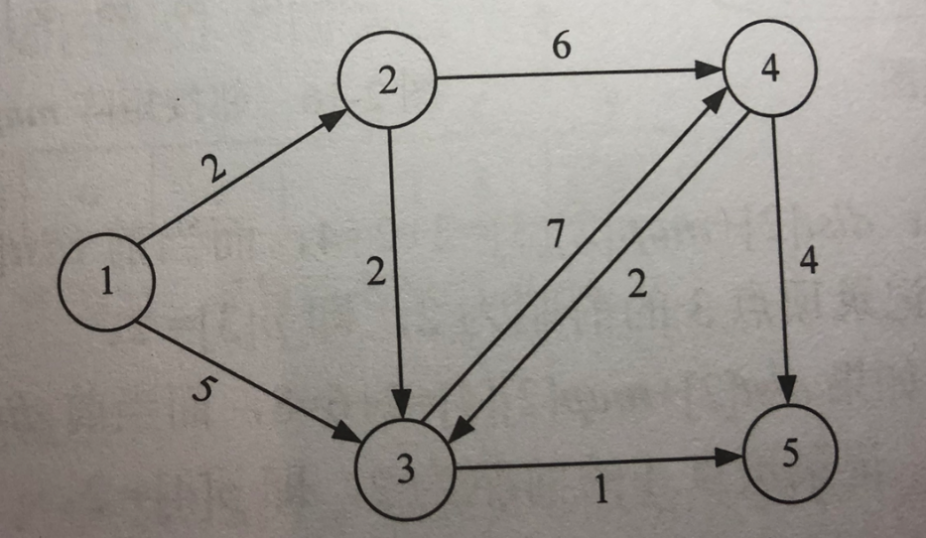
3: 65

**思路：从最轻的开始考虑，让最轻的和最重的一条船，若超出重量则可判定最**重的只能一人一条船

|  |
| --- |
| Task2.cpp |
| #include <bits/stdc++.h>  #define endl '\n'  using namespace std;  typedef long long ll;  void solve() {  ll n, c;  cin >> n >> c;  std::vector<ll> v(n);  std::vector<pair<ll, ll> > ans(n);  for (ll i = 0; i < n; i++)  cin >> v[i];  sort(v.begin(), v.end());  ll l = 0, r = n - 1, min = 0;  while (l < r) {  while (v[l] + v[r] > c && l < r) {  r--;  ans[min++] = {v[r], -1};  }  ans[min++] = {v[l++], v[r]};  }  cout << min << "\n";  for (ll i = 0; i < min; i++) {  // cout << i + 1 << ":" << ans[i].first << " " << ans[i].second == -1 ? "" : ans[i].second << "\n";  cout << i + 1 << ":" << ans[i].first;  if (ans[i].second != -1)  cout << " " << ans[i].second;  cout << "\n";  }  }  int main() {  int T;  // cin >> T;  T = 1;  while (T--) {  solve();  }  return 0;  } |

**3、单源最短路径**

现有一张景区地图，从1号结点出发，求到其他各个顶点的最短路径。



输出示例：

1-2：路径是1->2，长度为2

1-3：路径是1->2->3，长度为4

……..

提示：

dist[]=[0,2,4,8,5] //源点到其余各个顶点的最短路径长度

p[]=[-1,1,2,2,3] //前驱数组, 找出最短路径上经过的城市

|  |
| --- |
| Task3.cpp |
| #include <bits/stdc++.h>  #define endl '\n'  using namespace std;  typedef long long ll;  struct Edge  {  ll from\_node;  ll end\_node;  ll lenth;  };  void solve() {  int node\_num, edge\_num;  cin >> node\_num >> edge\_num;  std::vector<ll> dist(node\_num + 1, LLONG\_MAX);  std::vector<ll> pre(node\_num + 1, -1);  std::vector<Edge> edge(edge\_num);  dist[1] = 0;  for (ll i = 0; i < edge\_num; i++)  cin >> edge[i].from\_node >> edge[i].end\_node >> edge[i].lenth;  // Dijstra  priority\_queue<pair<ll, ll>, vector<pair<ll, ll>>, greater<pair<ll, ll>>> q;  q.push({0, 1});  while (!q.empty()) {  ll now\_dist = q.top().first, now\_node = q.top().second;  q.pop();  if (now\_dist > dist[now\_node]) continue;  for (ll i = 0; i < edge\_num; i++)  if (edge[i].from\_node == now\_node) {  if (dist[edge[i].end\_node] > dist[edge[i].from\_node] + edge[i].lenth) {  dist[edge[i].end\_node] = dist[edge[i].from\_node] + edge[i].lenth;  pre[edge[i].end\_node] = edge[i].from\_node;  q.push({dist[edge[i].end\_node], edge[i].end\_node});  }  }  }  cout << "Num:\t";  for (int i = 1; i <= node\_num; i++)  cout << i << "\t";  cout << "\nDist:\t";  for (int i = 1; i <= node\_num; i++)  cout << dist[i] << "\t";  cout << "\nPre:\t";  for (int i = 1; i <= node\_num; i++)  cout << pre[i] << "\t";  }  int main() {  int T;  // cin >> T;  T = 1;  while (T--) {  solve();  }  return 0;  } |

**\*4、乘船问题-升级版（选做题，POJ1700）**

**问题描述：**

某班级春游，组织水上游船活动，每艘游船限载两人。现有n个同学，每人划船熟练程度不同，过河时间分别为w1、w2、…、wn。假设只有一艘游船，游船过河时间为划船最慢同学的过河时间，游船需返回接送其他同学。请问最快需要多长时间才能保证班级所有同学都能过河。

**输入示例：(第一行为测试案例数量m，每个测试案例两行，第一行为学生人数n，第二行为每个学生的过河时间**w1、w2、…、wn**)**

1

4

1 2 5 10

**输出示例：(m行，每一行为一个测试案例的最少过河时间)**

17

**备注：这是一个竞赛中比较简单的贪心算法题，但容易出错，请选做的同学在POJ网站上提交一下试试看是否能够AC。**

**不会做的同学可以网上搜索解析，百度POJ1700即可。**

|  |
| --- |
| Task4.cpp |
| #include <bits/stdc++.h>  #define endl '\n'  using namespace std;  typedef long long ll;  void solve() {  ll n, sum = 0;  cin >> n;  std::vector<ll> v(n);  for (ll i = 0; i < n; i++)  cin >> v[i];  sort(v.begin(), v.end());  while (n >= 4) {  ll sum1 = 2 \* v[0] + v[n - 1] + v[n - 2];  ll sum2 = 2 \* v[1] + v[0] + v[n - 1];  if (sum1 > sum2)  sum += sum2;  else  sum += sum1;  n -= 2;  }  if (n == 3)  sum += v[0] + v[1] + v[2];  else if (n == 2)  sum += v[1];  else if (n == 1)  sum += v[0];  cout << sum << endl;  }  int main() {  int T;  cin >> T;  // T = 1;  while (T--) {  solve();  }  return 0;  } |