```cpp
#include<stdio.h>
#include<string.h>
#include<vector>
#include<utility>
#include<algorithm>
using namespace std;

#define MAX 1123456

typedef pair<int, int>ii;

int n, pai[MAX], ra[MAX];
vector< pair<int, ii> >LA;

void Make(int x) { pai[x] = x; ra[x] = 0; }
int Find(int i) { return (pai[i] == i) ? i : (pai[i] = Find(pai[i])); }
int isSameSet(int i, int j) { return Find(i) == Find(j);  }
void Union(int i, int j) {
  int x, y;
  if (!isSameSet(i, j)) {
    x = Find(i); y = Find(j);
    if (ra[x] > ra[y]) pai[y] = x;
    else {
      pai[x] = y;
      if (ra[x] == ra[y]) ra[y]++;
    }}}

int kruskal() {
  int i, ret = 0;
  for (i = 0; i < (int)LA.size(); i++) {
    pair<int, ii> front = LA[i];
    if (!isSameSet(front.second.first, front.second.second)) {
      ret += front.first;
      Union(front.second.first, front.second.second);
    }
  }
  return ret;
}


int main(void) {
  int i, u, v, w, menor, maior;
  scanf("%d", &n);
  for (i = 0; i < n; i++) {
    scanf("%d %d %d", &u, &v, &w); u--; v--;
    LA.push_back(make_pair(w, ii(u, v)));
  }
  for (i = 0; i < (int)LA.size(); i++) Make(i);
  sort(LA.begin(), LA.end());
  menor = kruskal();
  reverse(LA.begin(), LA.end());
  for (i = 0; i < (int)LA.size(); i++) Make(i);
  maior = kruskal();
  printf("%d\n%d\n", maior, menor);
  return 0;
}
```