Log Out Contact Help Center Q Search

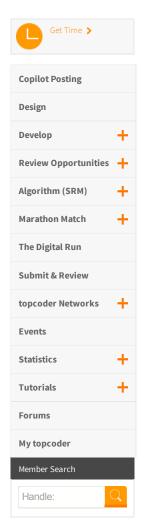


Challenges Cor

Community About topcoder

Blog

My Home



```
Statistics
                                                                                     Problem Statistics
   Problem Statistics for 2004 TCCC > Online Round 4 > Room 4 >
                                                                                                                                                                                                                                                                                                                                                    Difficulty
       [BadNeighbors]
                                                                                                                                                       maxDonations
                                                                                                                                                                                                                                                                                                                                                   Level One
      BombMan
                                                                                                                                                         shortestPath
                                                                                                                                                                                                                                                      problem details
                                                                                                                                                                                                                                                                                                                                                  Level Two
                                                                                                                                                             warnings
      ScriptLanguage
                                                                                                                                                                                                                                                      problem details
                                                                                                                                                                                                                                                                                                                                                 Level Three
                                                                                                                                                                                                                                         View BadNeighbors Problem Statement
  #include <cmath>
  #include <iostream>
 #include <set>
#include <string>
  #include <vector>
  // Macros
 typedef long double ld;
 typeder long long l1;

template <class T> T MIN(const T &x, const T &y) {return x<y? x:y;}

template <class T> T MAX(const T &x, const T &y) {return x>y? x:y;}

template <class T> T ABS(const T &x) {return x>0? x:-x;}

template <class T> void SWAP(T &x, T &y) {T z=x; x=y; y=z;}
 const 1d EPSILON = 0.000000000001;
 If toUnit(String S) (stringstream in(s, ios_base::in); ld result; in >> result; return result;)

Id toDouble(string s) (stringstream in(s, ios_base::in); ld result; in >> result; return result;)

string toString(ll n) (stringstream out(ios_base::out); out << n; return out.str();}

string toString(ld d, int precision = -1) (stringstream out(ios_base::out); if (precision >= 0) (out.precision(precision); out

bool nonZero=false; for(int i=0;i<s.size();i++) if (s[i]!='0' && s[i]!='-' && s[i]!='.') nonZero=true; if (s[0]=='-' && !non:

string toString(int n) {return toString((ll)(n));}
string toString(const string &s) {return s;}
template <class U, class V> string toString(pair<U,V> u) {return toString("(") + toString(u.first) + "," + toString(u.second) template <class T> string toString(const vector<T> &v, string delim=",") {string s="("; for(int i=0;i<v.size();i++) {if(i!=0)stemplate <class T> string toString(const set<T> &t, string delim=",") {string s="
{"; for(typename set<T>::const_iterator it=t.begin(); it!=t.end(); it!+) {if(it!=t.begin())s!=delim+" "; s += toString(*it);} template <class U, class V> string toString(const map<U,V> &m, string delim=",") {string s="
{"; for(typename map<U,V>::const_iterator it=m.begin(); it!=m.end(); it+) {if(it!=m.begin())s!=delim+" "; s += toString(it->f. int getNumWords(string s, string delim=" ") {int count=0; for(int i=0; i<s.size(); i++) if ( (i==0 || delim,find(s[i-1])!=-1) string getWord(string s, int j, string delim=" ") {int count=0; string t; for(int i=0; i<s.size(); i++) {if (delim.find(s[i])=1) = -1) count++; if(count>j) return t; t="";}} return t;
string getDelimiter(string s, int j, string delim=" ") {int count=0; string t; for(int i=0; i<s.size(); i++) {if (delim.find(s[i])=1) | i= -1) count++; if(count>j) return t; t="";}} return t;
bool isEqual(ld v1, ld v2) {return ABS(v1-v2) < EPSILON;}
bool isLess(ld v1, ld v2) {return v1-v2 < EPSILON;}
ll gcd(ll n1, ll n2) {return n2==0? ABS(n1) : gcd(n2,n1%n2);}
ll lcm(ll n1, ll n2) {return n1==0 && n2==0? 0 : ABS(n1*n2)/gcd(n1,n2);}
   // Real code
  class BadNeighbors
        vector<int> blah;
         int getAnswer(int i1, int i2)
             int key = i1*500+i2;
             if (i1 > i2) return 0;
answer[key] = MAX(getAnswer(i1+1,i2), getAnswer(i1+2,i2)+blah[i1]);
              return answer[key];
         int maxDonations(vector <int> donations)
  System Test Results
```

```
{11, 15}
{7,7,7,7,7,7,7}
{94, 40, 49, 65, 21, 21, 106, 80, 92, 81, 679, 4, 61, 6, 237, 12, 72, 74, 29, 95, 265, 35, 47, 1, 61, 397, 52, 72, 37, 51, 1, 81, 45, 435, 7, 36, 57, 86, 81, 72}
{347, 49, 608, 460, 67, 856, 21, 526, 552, 412, 761,
286, 481, 441, 598, 933, 462, 328, 92}
{54, 161, 763, 454, 217, 338, 313, 812, 72, 634, 764,
595, 523, 844, 65, 1000, 292}
\{374,429,448,480,751,781,295,920,442\}
{836, 100, 274, 931, 632, 817, 951, 992, 17, 349, 189,
99, 77, 316, 698, 10, 732, 3, 2, 445, 524, 343, 953,
{596, 633, 339, 994, 719, 901, 551, 772, 38, 211, 212,
812, 454, 931, 879, 78, 243, 637, 815, 182, 359, 192,
892, 152, 234, 480, 814, 649, 613, 216}
{15, 992, 551, 28, 245, 149, 647, 633, 422, 137, 3,
805, 177, 848, 79, 878, 482}
{610, 467, 596, 278, 889, 334, 886, 721, 842, 708,
610, 491, 985, 995, 219, 710, 78, 267, 887, 404, 152,
372, 127, 152, 915}
{399, 470, 76, 788, 170, 349, 895, 682, 49, 873, 943,
798, 111, 981, 678, 600, 459, 265, 939, 555, 406}
{874, 173, 392, 916, 119, 997, 923, 841, 301, 755,
{35, 991, 80, 273, 637, 811, 974, 845, 725}
{593, 351, 111, 474, 913, 520, 468, 738, 561, 122,
568, 808, 374, 918, 907, 349, 110, 320, 448, 772, 921}
{13, 804, 513, 308, 752, 771, 237, 769, 945, 932, 858, 171, 462, 739, 627, 797, 172, 662, 589, 954, 488, 92, 788, 557, 1000, 881, 107, 697, 849, 265, 28}
\{110, 310, 520, 631, 927, 481, 595, 38, 574, 651, 799,
79, 933, 659, 69, 418, 93, 527, 929, 938, 920, 580,
960, 808, 454, 989, 832, 775, 985, 738, 973, 13, 434,
603}
{918, 501, 486, 579, 964, 638, 396, 988, 85, 503}
{848, 106, 26, 171, 771, 401, 587, 6, 711, 318, 93,
964, 3, 893, 942, 204, 57, 51, 958, 665, 1, 767, 370}
{380, 659, 849, 70, 280, 85, 408, 386, 860, 113, 708}
{90, 1, 2, 89, 3, 4, 88, 5, 6, 87, 5, 4, 86, 3, 2, 85, 1, 2}
{90, 1, 2, 89, 3, 4, 88, 5, 6, 87, 5, 4, 86, 3, 2, 85}
{162, 964, 975, 847, 31, 472, 61, 903, 713, 392, 798,
937, 378, 994, 969, 310, 924, 766, 529, 422, 786, 844,
746, 428, 332, 832, 280, 233, 609, 744, 165, 635, 335,
655, 414, 661, 686, 374, 739, 599}
{265, 40, 953, 368, 357, 880, 302, 970, 349, 91, 841,
894, 707, 434, 713, 117, 108, 524, 429, 275, 817, 225, 557, 311, 795, 24, 895, 190, 776, 410, 534, 207, 675,
705, 652, 295, 777, 363, 808, 255}
{148, 496, 796, 25, 538, 254, 819, 942, 885, 760, 878,
450, 202, 463, 775, 853, 202, 168, 49, 532, 573, 482,
528, 508, 176, 103, 886, 348, 792, 198, 746, 193, 239,
102, 452, 132, 837, 693, 546, 219}
{221, 118, 376, 194, 434, 965, 42, 347, 109, 611, 22,
72, 270, 213, 459, 305, 982, 749, 30, 481, 598, 578, 744, 625, 882, 224, 800, 896, 612, 480, 862, 250, 596,
464, 700, 859, 198, 962, 187, 618}
{590, 260, 60, 331, 929, 661, 675, 181, 593, 69, 253, 675, 410, 328, 734, 560, 547, 520, 408, 108, 346, 14, 982, 808, 944, 896, 558, 691, 941, 341, 306, 471, 750,
592, 279, 205, 485, 483, 895, 415}
{442,851,111,479,261,770,194,619,864,263,
323, 15, 110, 37, 926, 399, 916, 824, 811, 988, 290, 135, 633, 766, 293, 554, 193, 943, 480, 239, 391, 865,
778, 962, 383, 792, 42, 836, 646, 727}
{821, 859, 221, 922, 912, 563, 326, 764, 394, 308,
934, 509, 725, 267, 665, 727, 352, 706, 494, 366, 44, 942, 422, 91, 804, 300, 168, 901, 800, 432, 406, 54, 216, 621, 818, 36, 311, 56, 487, 629}
{251, 636, 734, 142, 493, 533, 532, 519, 566, 617, 81,
344, 125, 862, 919, 742, 66, 3, 224, 770, 275, 55, 574,
210, 355, 785, 513, 571, 266, 855, 701, 720, 364, 163,
314, 809, 780, 843, 622, 57}
{397,510,335,521,973,576,420,180,89,910,554,570,702,654,290,685,6,786,302,91,249,851,799,9,879,387,292,628,129,714,449,78,717,
159, 650, 281, 869, 271, 253, 364}
{268, 265, 254, 470, 850, 746, 225, 541, 595, 773,
924, 112, 597, 413, 965, 222, 722, 287, 263, 326, 109,
4, 776, 458, 621, 34, 211, 982, 108, 295, 974, 978, 1,
677,600,671,318,779,328,358}
{659, 384, 265, 947, 121, 568, 795, 596, 361, 906,
517, 902, 233, 906, 989, 777, 511, 694, 21, 422, 232, 956, 48, 810, 80, 328, 387, 694, 77, 748, 828, 422, 546, 764, 15, 810, 961, 423, 608, 879}
{821, 179, 5, 532, 129, 182, 720, 138, 418, 438, 182,
564, 506, 581, 784, 386, 327, 432, 526, 914, 265, 58, 475, 171, 363, 835, 568, 161, 694, 845, 2, 592, 945,
23, 555, 735, 958, 658, 715, 236}
```

```
577, 193, 420, 514, 654, 811, 522, 843, 485, 940,
877, 345, 234, 639, 42, 885, 178, 820, 178, 304, 687,
388, 666, 525, 907, 666, 335, 783, 83, 960, 958, 672,
252, 668, 864, 739, 842, 155, 67, 67}
{349,500,347,682,836,308,69,628,472,10,359,
238, 707, 467, 439, 500, 974, 229, 573, 362, 987, 501,
45, 523, 813, 741, 78, 543, 903, 792, 369, 65, 161,
667, 225, 581, 977, 388, 862, 443}
{835, 784, 541, 755, 778, 788, 621, 371, 521, 561,
519, 482, 442, 727, 803, 158, 59, 112, 334, 792, 824,
622, 465, 602, 245, 355, 777, 197, 403, 567, 738, 793,
918, 884, 972, 786, 284, 22, 529, 280}
{291, 322, 665, 641, 930, 599, 164, 305, 794, 193,
145, 886, 477, 571, 658, 74, 892, 553, 85, 971, 747,
15, 524, 253, 634, 930, 872, 5, 326, 735, 310, 163,
909.317.206.232.310.338.286.372}
{653, 257, 39, 982, 596, 137, 954, 645, 501, 277, 139,
667, 825, 750, 302, 452, 610, 195, 837, 673, 332, 560, 1000, 504, 644, 75, 760, 210, 738, 409, 927, 10, 176,
831, 592, 468, 15, 196, 744, 621}
{966, 778, 174, 654, 174, 248, 411, 267, 880, 624,
490, 737, 911, 628, 832, 339, 425, 686, 330, 995, 834,
766, 63, 930, 135, 344, 700, 690, 143, 54, 743, 82,
313, 5, 267, 448, 853, 309, 685, 264}
{846, 103, 386, 181, 956, 244, 797, 514, 97, 883, 523,
642, 106, 336, 766, 975, 105, 19, 116, 921, 700, 745,
172, 763, 587, 309, 173, 492, 482, 173, 960, 390, 466,
980, 166, 625, 9, 811, 453, 185}
{940, 374, 16, 245, 610, 761, 321, 435, 296, 25, 219
585, 697, 586, 190, 184, 993, 401, 416, 198, 539, 883, 815, 918, 717, 838, 958, 168, 858, 11, 984, 578, 429, 426, 269, 601, 879, 388, 599, 612}
\{622, 863, 50, 735, 668, 980, 585, 25, 279, 560, 634,
228, 900, 898, 570, 636, 283, 514, 33, 493, 401, 228,
512, 990, 248, 697, 492, 124, 84, 505, 682, 641, 28,
19, 586, 771, 148, 463, 700, 764}
{786, 120, 276, 229, 656, 377, 478, 927, 852, 9<u>1, 128,</u>
341, 206, 564, 910, 736, 1, 661, 157, 900, 132, 990, 960, 786, 605, 517, 517, 630, 762, 739}
{998, 41, 740, 633, 630, 927, 358, 887, 639, 209, 445,
374, 619, 616, 659, 632, 500, 390, 558, 722, 486, 392,
791, 928, 308, 873, 50, 409, 71, 153, 50, 458, 410, 45}
{307, 322, 35, 140, 667, 1, 884, 44, 546, 239, 548,
881, 222, 459, 55, 158, 642, 280, 237, 561, 925, 539,
38, 699, 107, 116, 449, 418, 720, 487, 983, 743}
{917, 19, 978, 687, 346, 245, 677, 708, 565, 940, 211,
127, 993, 768, 469, 279, 460, 335, 734, 130, 691, 783
182, 391, 827, 295, 534, 263, 193, 498, 327, 120, 690}
{930, 235, 680, 718, 932, 471, 908, 14, 754, 695, 76,
601, 388, 606, 957, 580, 961, 512, 510, 979, 442, 602,
341, 641, 153, 196, 974, 937, 936, 487, 820, 243, 738,
313}
{599, 122, 578, 834, 875, 684, 43, 120, 304, 668, 346,
820, 247, 109, 246, 174, 615, 696, 342, 292, 458, 168,
671, 725, 856, 451, 914, 880, 385, 961, 187, 630, 121,
\{353, 440, 165, 53, 928, 155, 362, 835, 399, 526, 387,
444, 992, 721, 430, 493, 708, 927, 112, 997, 772, 49,
60, 586, 20, 363, 182, 866, 762, 378, 666, 544, 98,
175, 290, 665, 703}
{305, 392, 546, 548, 795, 60, 160, 392, 107, 871, 565,
810, 708, 463, 157, 810, 535, 783, 454, 541, 735, 129,
813, 394, 492, 973, 927, 450, 496, 189, 160, 695}
{606, 549, 257, 305, 395, 604, 477, 488, 643, 885, 46,
141, 626, 699, 563, 916, 696, 145, 814, 689, 430, 191, 826, 259, 641, 414, 180, 569, 140, 514, 432, 89, 682,
781, 491}
{349, 187, 672, 268, 750, 926, 787, 578, 784, 795,
991, 9, 416, 981, 355, 840, 850, 501, 53, 794, 712
740, 833, 113, 401, 285, 952, 422, 535, 532, 102
{505, 335, 897, 676, 535, 315, 426, 899, 240, 304,
986, 370, 648, 816, 572, 737, 818, 349, 916, 778, 689, 785, 872, 909, 562, 172, 710, 702, 658, 665, 465, 94,
522, 476, 828, 677, 674}
{172, 190, 766, 550, 250, 69, 703, 254, 746, 730, 638,
908, 379, 74, 470, 473, 457, 529, 505, 186, 865, 873,
239, 528, 238, 958, 788, 731, 695, 428, 360, 69}
{473, 313, 573, 144, 708, 739, 673, 913, 268, 258,
729, 444, 310, 647, 367, 384, 193, 239, 19, 259, 761,
531, 106, 368, 306, 319, 847, 317, 983, 729, 771, 69,
695, 973, 626, 797, 689, 258, 516}
{445, 298, 806, 321, 703, 71, 514, 289, 881, 992, 675,
644, 470, 469, 430, 353, 461, 907, 392, 394, 244, 544, 185, 350, 958, 192, 930, 453, 726, 844, 819, 197, 797, 323, 889, 835, 142, 650, 538, 87}
{544, 343, 159, 19, 538, 237, 747, 936, 522, 189, 47,
167, 123, 88, 891, 10, 232, 674, 77, 617, 162, 995,
614, 356, 369, 47, 100, 515, 408, 988, 676, 456, 816,
523, 56, 236, 891}
\{924, 602, 959, 7, 608, 218, 523, 697, 292, 980, 881,
265, 205, 256, 198, 862, 142, 940, 715, 482, 425, 288, 791, 929, 876, 594, 704, 404, 682, 901, 719, 450, 81,
114, 738, 498, 657, 72, 966}
{645, 358, 416, 829, 50, 903, 750, 931, 600, 705, 910,
```

593, 487, 337, 221, 789, 630, 227, 171, 488, 702, 131, 867, 260, 501, 580, 255, 385, 319, 573, 527, 664, 854, 236]
{965, 850, 698, 178, 936, 112, 944, 46, 288, 741, 23, 903, 454, 448, 539, 578, 469, 579, 32, 703, 424, 61, 488, 178, 902, 797, 933, 55, 380, 209, 791, 226, 739, 474, 431, 388, 614, 745}
{281, 549, 86, 524, 881, 611, 104, 384, 274, 353, 302, 581, 237, 805, 988, 874, 100, 482, 166, 190, 363, 206, 206, 21, 659, 945, 273, 554, 84, 740, 392}
{617, 244, 782, 414, 340, 346, 583, 148, 435, 840, 360, 773, 488, 160, 873, 751, 438, 695, 410, 910, 652, 350, 617, 727, 970, 596, 184, 452, 256, 408, 903, 207}

