Zadanie 1 [15 punktów]

W tym zadaniu badamy koła o promieniach **jednostkowych** całkowicie zawartych w **pierwszej ćwiartce** kartezjańskiego układu współrzędnych i o środkach w punktach o współrzędnych całkowitoliczbowych.

a) **[5 punktów]** Zaproponuj strukturę danych dla dynamicznego, skończonego zbioru kół *S* umożliwiającą wydajne wykonywanie następujących operacji:

   Ini:: zainicjuj zbiór *S* jako pusty – operacja wykonywana tylko raz na samym początku;

   Insert(a,b):: dodaj do *S* koło o środku w *(a,b)*, gdzie *a*, *b* dodatnie liczby całkowite;

   Delete(a,b):: usuń z S koło o środku w *(a,b)*, gdzie *a*, *b* dodatnie liczby całkowite;

   Find(x,y):: podaj do ilu kół w zbiorze *S* należy punkt o współrzędnych rzeczywistych *(x,y)*.

b) **[10 punktów]** Dany jest zbiór *S* zawierający *n* kół o współrzędnych mniejszych od n^2 oraz para punktów o współrzędnych rzeczywistych $P=(x_1,y_1)$, $Q=(x_2,y_2)$. Zaproponuj efektywny algorytm, który sprawdzi, czy punkty *P* i *Q* można połączyć krzywą całkowicie zawartą w sumie teoriomnogościowej kół ze zbioru *S*.

Zadanie 2 [10 punktów]

Zaprojektuj strukturę danych dla dynamicznego, *n*-elementowego multizbioru liczb naturalnych *S* umożliwiającą wydajne wykonywanie następujących operacji:
Ini:: *S* := {0,0, …, 0} – zainicjuj S z n zerami;
Inc(k):: wybierz k najmniejszych liczb w zbiorze i każdą z nich zwiększ o 1, 1<= k <= n;
Max:: podaj wartość największego elementu w zbiorze;
Min:: podaj wartość najmniejszego elementu w zbiorze.

Zadanie 3 [15 punktów]

Dany jest ciąg *A*, *n* parami różnych liczb całkowitych $a_1, a_2, \ldots, a_n$ oraz kształt drzewa poszukiwań binarnych T powstającego w wyniku wstawienia do początkowo pustego drzewa kolejno liczb $a_1, a_2, \ldots, a_n$. Czy informacja o kształcie drzewa *T* pomaga w posortowaniu liczb $a_1, a_2, \ldots, a_n$? Załóżmy, że o *T* wiemy tylko, że jest drzewem o wysokości *n*-1.

a) **[2 punkty]** Udowodnij, że posortowanie *A* wymaga w pesymistycznym przypadku, co najmniej *n*-1 porównań pomiędzy elementami ciągu.

b) **[3 punkty]** Zaproponuj algorytm sortowania *A* wykonujący, co najwyżej *n*-1 porównań pomiędzy elementami ciągu.

c) **[5 punktów]** Zaproponuj optymalny algorytm znajdujący przez porównania maksimum w ciągu *A*.

d) **[5 punktów]** Załóżmy, że *n=2k*, dla pewnego, całkowitego *k* > 0, a o *T* wiemy tylko, że jest drzewem, w którym skrajnie prawa ścieżka składa się z dokładnie k węzłów, z których każdy posiada lewego syna. Udowodnij, że w tym przypadku do posortowania *A* potrzeba w pesymistycznym przypadku $\Omega(n\log n)$ porównań.