CS 4820, Spring 2014	**Homework 4, Problem 2**
Name: Piyush Maheshwari	
NetID: pm489	
Collaborators: None	

(2) Solve Chapter 5, Exercise 7 in Kleinberg & Tardos.

Solution

Given any n x n grid graph. Let e_{ij} be the node which corresponds to the ordered pair (i,j) in the graph. Let the set of elements B define boundary of the graph. This means the set B contains

B = $\{e_{1j} \forall j = 1..n\} \cup \{e_{nj} \forall j = 1..n\} \cup \{e_{j1} \forall j = 1..n\} \cup \{e_{jn} \forall j = 1..n\}$

Every element of the set B would only be connected to one node which is not in B. The remaining neighbours will be in B. This is by how the grid graph is defined in the question. Let b_{min} be the minimum element in the boundary set B of a grid graph G. Let x be the node to which is b_{min} is connected to such that x $\notin$ B. Clearly there will be only one such element as mentioned above. Let x be called the inner neighbour of b_{min}. Clearly if $b_{min} < x$, then b_{min} will be the local minimum. Also $b_{min} \neq x$ because all nodes have been assigned distinct weights.

Lemma 1. *If $b_{min} > x$, then there must be a local minimum within a reduced graph G' with node set nodes(G) - B.*

Proof. We will prove this by induction. We claim that the statement above holds true for any n.

<u>Base Case</u>
We will have our base case as n = 3. Any graph for less than n = 3 would not have any elements left with we remove its outer boundary. Let b_{min} be the min element of the boundary B of the graph G of size n = 3. After removing B we will only have a single element x in the graph G'. Now if $x < b_{max}$, then x is the local minimum since x is smaller than all its neighbours.

<u>Inductive Step</u>
Consider graph G of size n+1 * n+1. Let b_{min} be the minimum element of the boundary B of the G. Let x which is the inner neighbour of b_{min} has neighbours a,b,c. If $x < a$ and $x < b$ and $x < c$, then x is the local minimum since we know b_{min} > x and the induction holds. Otherwise find the min element $b'_{min}$of the outer boundary set B$'$ of the graph G$'$ which is obtained by removing outer boundary of G. If the inner neighbour of b'_{min} which is e', is greater than b'_{min}, then b'_{min} is the local minima. Otherwise using the inductive hypothesis we can say that there will be a local minima in G'' which is formed by removing B' from G' since G' is of size n x n. Hence the induction holds. □

Now consider a graph G of size n x n. Consider the following set of nodes

M = $\{e_{1j} \forall j = 1..n\} \cup \{e_{nj} \forall j = 1..n\} \cup \{e_{j1} \forall j = 1..n\} \cup \{e_{jn} \forall j = 1..n\} \cup \{e_{jn/2} \forall j = 1..n\} \cup \{e_{n/2j} \forall j = 1..n\}$

M is essentially the set of nodes which form the boundary of G plus all the nodes on $n/2^{th}$ row and all the nodes of $n/2^{th}$ column.

Now consider the center element of the graph G which is situated at (n/2,n/2). If this element is less than all its four neighbours then it is the local minima. Otherwise find the minimum element m_{min} in set M of G. Now there are two cases :

1. $\underline{m_{min} \text{ lies on the outer boundary}}$

   WLOG assume that m_{min} lies on the node (1, i) with $i > n/2$ which is the topmost row. This means m_{min} lies in first quadrant boundary. Other cases will be completely symmetrical. In this case m_{min} would have one neighbour which is not in M. Let that neighbour be e. This neighbour would lie within the subgraph G' formed by considering nodes between the rectangle (2,2) and (n/2-1, n/2-1) which is also the first quadrant of the graph visualizing the graph as a cartesian space with the origin at n/2,n/2.

   Now if $e > m_{min}$ then m_{min} is the local minima. Other wise from lemma 1 we know that a local minima would lie in the sub graph G′ since we have considered the boundary covering the graph G' while considering the set of elements in M. Hence we can recursively solve the subproblem G′ which is of size n/2 x n/2.

2. $\underline{m_{min} \text{ lies either on the row n/2 or the column n/2 and not on boundary}}$

   In this case WLOG assume that m_{min} lies on (n/2,i) $i > n/2$ since all other cases are exactly symmetric and can be argued in a similar way. Now m_{min} would have two neighbours which are not in M, one of which lies in the first quadrant (a) and one of which lies in the fourth quadrant (b). Now if $m_{min} < a$ and $m_{min} < b$, then m_{min} is the local minima and we are done. Otherwise choose any value a or b which is smaller than m_{min} and recurse in that corresponding quadrant since we know from lemma 1 that we will have a minima in that quadrant. This is simple to see because we have found a element which is less than the min boundary element which is exactly what lemma 1 says. Again we have a sub problem of size n/2 X n/2.

.

Lemma 2. *Every grid graph having distinct values on its nodes will have a local minima*

Proof. This follows from the discussion above and lemma 1. If $b_{min} < x$, then b_{min} will be the local minimum, otherwise from lemma 1 we will have a local minima in graph G'. □

So the algorithm can be summarized as follows-

Algorithm 1

procedure LOCAL-MINIMA(n x n)
    if element at n/2, n/2 is local minima **then**
        return the element
    else
        Else calculate the set M as described above
        Find m_{min}
        if Check if m_{min} is the local minima **then**
            return the element
        else
            Check the neighbours of m_{min} as described above based on where m_{min} lies.
            Find the correct quadrant to recurse
            return local-minima(n/2,n/2)

Running Time

First lets calculate the running time of the algorithm.

Assume the graph contains n nodes in total. The number of nodes along a edge are $\sqrt{n}$. We spend O($\sqrt{n}$) time in making the set M and finding m_{min} since we make constant number of passes over an edge of the grid. And if needed at each step we solve a smaller subproblem of size n/4 since we only choose a single quadrant out of four.

The recurrence is

T(n) = T(n/4) + O($\sqrt{n}$)
Applying master theorem in this case. a = 1, b = 4 and c = 1/2.
Since $c > log_b(a)$, we have T(n) = O(n^c) = O($n^{1/2}$).

In the original problem we have a n x n grid, hence the input size is n^2, hence the running time is T(n) = O($n^{2(1/2)}$) = O(n)

Proof of correctness

Proof of correctness directly follows from lemma 1. In each call to the function we check if the mid point is the local minima. Otherwise we check if the m_{min} is the local minima. Otherwise we know from lemma 1 that a local minima would lie in the quadrant that we have chosen and we solve that sub problem.

Now we know that this algorithm will find a local minima since we know from lemma 2 that a local minima would always exist. Now we have to prove that we have used O(n) probes for a n x n grid.

Consider a call to a method with grid of size n x n. The number of probes used in finding m_{min} is O(n) since we do a constant number of passes over the edge whose size is n. Otherwise we solve it recursively.

Hence the total number of probes will be T(n x n) = T((n/2 x n/2) /4) + O(n).
This is exactly similar to the recurrence that we solved above for getting the running time of the algorithm . Hence T(n x n) = O(n). This means that we have solved this problem in O(n) probes.