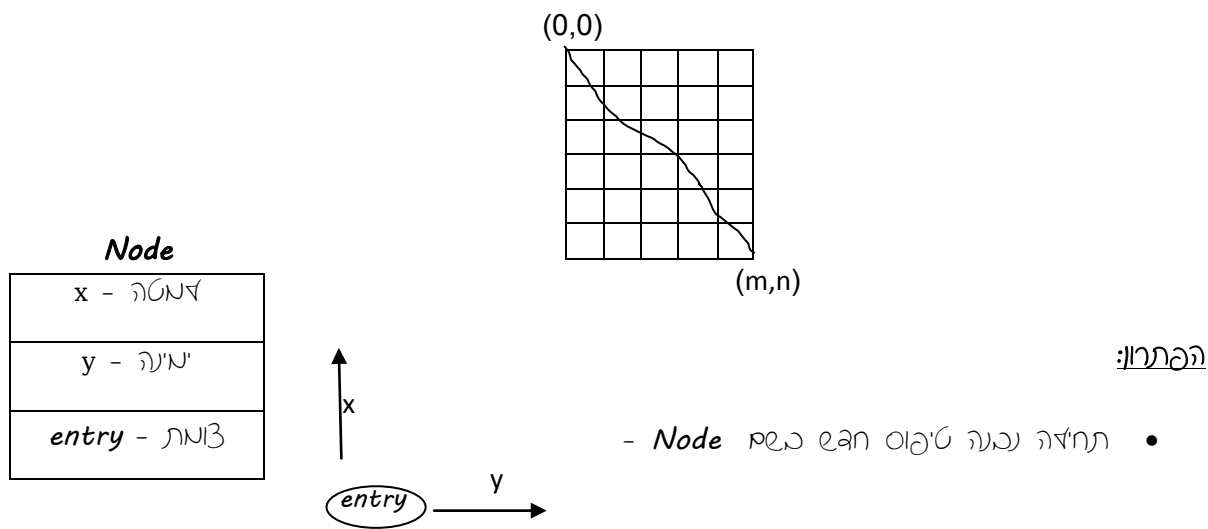


בעיית המטוס

הבעיה: צמצום אלגוריתם המוצא את המסלול הקצר או הארוך ביותר מנקודה (0,0) ל-(m,n):



- 1. נבנה מטרצה של **Node**-ים עם ערכים כגון $y-1$.
- 2. טיפוס את צומת (0,0).
- 3. שותחל שורה וטור ראשוניים דפ'.

```
mat[0][i].entry = mat[0][i-1].entry + mat[0][i-1].x  
mat[i][0].entry = mat[i-1][0].y + mat[i-1][0].entry
```

4. נמצאו את כל שאר המטרצה דפ' הכללית:

```
x = mat[i-1][j].entry + mat[i-1][j].y  
y = mat[i][j-1].entry + mat[i][j-1].x  
mat[i][j].entry = Math.min ( x , y )
```

5. $mat[size-1][size-1].entry$ - זה הסכום המסלול הנמוך ביותר.

6. אם חוצים דהמשק ודחשכ יש את אחת המסלולים:

טיפוס 2 אינקקסיס דפיה הימנית התחתונה (סעיף 5), ומשם נסאוד כל פעם כשטיע צצומת- מאיפה הזעע, וכך נע דחשכ את המסלול. עק שטיע דנקובת (0,0).

סיכומים:

- חישוב המטרצה: $O(n * m)$.
- חישוב מסלול אחת: $O(n + m)$.

```

public class CheapestPath {
    int size; // גודל המטריצה
    Node[][] mat;

    public CheapestPath(int n){ // בנאי
        this.size = n;
        mat = new Node[n][n];
    }

    private class Node{ // מחזיקה נסתרת
        int y, x, entry;
        public Node(int x, int y){
            this.x = x;
            this.y = y;
            this.entry = 0;
        }
    }

    public int priceOfCheapestPaths(){ // חישוב המטריצה
        mat[0][0].entry = 0;
        for (int i=1; i<size; i++){ // אינרסיה
            mat[0][i].entry=mat[0][i-1].entry + mat[i][0].x;
            mat[i][0].entry= mat[i-1][0].y + mat[i-1][0].entry;
        }
        int x,y;
        for (int i=1; i<size; i++){ // הכחד דפ' מילוי
            for (int j=1; j<size; j++){
                x = mat[i-1][j].entry+mat[i-1][j].y;
                y = mat[i][j-1].entry+mat[i][j-1].x;
                mat[i][j].entry = Math.min(x,y);
            }
        }
        return mat[size-1][size-1].entry;
    }

    public void printPath(){ // חישוב אונת המסלולים
        String p = "("+size+","+size+")";
        int i = size-1, j = size-1;
        while (i>0 || j>0){
            if (i>0){
                int x = mat[i-1][j].entry+mat[i-1][j].y;
                if (mat[i][j].entry == x) i--;
                else j--;
            }
            else j--;
            p = p + "->("+i+1+","+j+1+")";
        }
        System.out.println("\nthe cheapest path: \n"+p);
    }
}

```