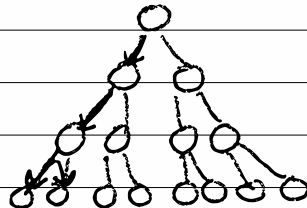


18. Find the maximum total from top to bottom of the triangle below:

75
95 64
17 47 82
18 35 87 10
: : : : : : : :
: : : : : : : :
: : : : : : : :
: : : : : : : :
: : : : : : : :
: : : : : : : :
: : : : : : : :
: : : : : : : :

If it was a binary tree, you could do a depth first search recursively



and return the current value with the max of the left & right recursive calls. It's a pyramid not a tree, but the same idea can be done: start at the second to last row, loop through values and add to each the max of the left & right child values.

75 75 75
 95 64 → 95 64 → 95 64
 17 47 82 52 134 82 52 134 169
 18 35 87 10 18 35 87 10 18 35 87 10

75 75
 229 64 → 229 233
 52 134 169 52 134 169

308
 229 233

```

triangle = [[75],
             [95, 64],
             [17, 47, 82],
             ;
  
```

```

def max_sum(t)
  (t.length-2).downto(0) do |i|
    t[i].each_with_index do |num, num-i|
      t[i][num-i] += t[i+1][num-i],
                    t[i+1][num-i+1]].max
    end
  end
  t[0][0]
end
  
```

