backtraking

February 19, 2020

1 BackTracking algorith

> Backtracking is an algorithmic-technique for solving problems recursively by trying to build a solution incrementally, one piece at a time, removing those solutions that fail to satisfy the constraints of the problem at any point of time (by time, here, is referred to the time elapsed till reaching any level of the search tree).

Often we can find backtracking and Recursive tecnique together!

Example: Give an integer "n". Print all the possible paris of n balaced parentheses. The output strings should be printed in the sorted order considering '(' has higher value than ')'.

~~~ def genParenthesis(openB, closeB, n, s=[]):

```
#Base Case
if(closeB == n):
    print(s)
    return
else:
    if(openB > closeB):
        #can definitely put one closing bracket
        s.append(")")
        genParenthesis(openB, closeB+1, n, s)
        s.pop() #remove last item in the list
    if(openB < n):
        s.append("(")
        genParenthesis(openB+1, closeB, n, s)
        s.pop()
return
```

> Elementi dell'algoritmo di backtracking: * OUR CHOICE * OUR CONSTRAINTS * OUR GOAL

1.1 our *CHOISE*

place a openB or closeB

1.2 our *CONSTRAINTS*

we cannot close until we open a bracket
~~~

1.3 our *GOAL*

n * 2 placements

Here a video explane: backtracking for dummies

[ ]: