## Dynamic Perogramming #) Edit Distance Given & stees & and t. Win. number of operations that need to be done to convert & to the time will 0 dresent @ Remove @ Replace. int l=0, se n#m; S= "--- tts" onseit sin s. (m, n-1) ((0, m) (n) salvav (1) be trav Replace bins Reque for (m-1, n-1) ( batise bing bin Remove 'b' in & Re au fou (1-m][1-n] bosidu) int dp [s. length()+1] [t.length+1] for (int i=0; ixm+1; i++) d suture si for (int j=0; j<n+1; j++)& → y(i==0) 11 s. empty. → onscut all chan of dp[i][]=j; -> else ef (j==0). Il Remove all chaus jaion s as t is empty. المنفلافط إن آل المنفقة المالية المنفقة المالية المنفقة المنفق → alse y (8[î-1] == t[j-1]) 11No ops. de[[][]= de[[-][-]];=(][]) -> else de [i] = 1+ min (de [i-0] -i], min (de [i-1]), γ of & a o & ( i, j-1, n, m, mid, guid, vilited): \$ retuin op Em [7]; Die min 14%, 3 ) x0 i sto

A)0

Scanned with CamScanner

```
# Maximize Cut sogmente would be to the second
  lut into 3 segments x, y ou z. Sotal no. of cut seg.
  must be maximum
               1000 100 100 0/P:- 41 14
 JIP:- K= H
     x= Q, y=1, Z=1.
      1) Make sure after out of foutsular length
      und length & >=0. (n-x) >,0
                          5 = 3110 -1 (n-y), ≥ 0 196 +1
      @ also make sure
          the cut is one going to make is available
     of [n-x] = (1).
                                   30 == m) 10 0319
     vector Lines of (n+1,-1);
     op[0]=0;
     for (int i=1; ix (n+1); i++) &
        1 Ci-x>0 22 op [i= x) [= -1) (i) mi) mi
            dp[e] = max(dp[e], 1+ dp[e-x));
        11 2 move y condit for y and x12
               :[][]] == mat [[][]];
    Z
    veluen (dp[n] <= 0) 2 0 : dp[n];
                     else of (f== (111-1))
              : Cillistem = Cilliste
[](1) = ] = 1 - mar (1-7) (1) (1) mar -1 = (1) (1) po
             (FF [F] 70, 200) con = 200
```

July mulling

```
# Maximum size Sq. - Submateux of 13
d|P:- [0100]
                      0/P:- 2'
                      O Risest lost now Z beep mater
                          else
                          I+ min (down, & night,
                                     diagonally down)
    int aptinj[m]; int ans=0;
    y (n==1) s
        for(j=0; j<m; j++) ans=max (ans, mat []][]);
   else \frac{1}{2} (m==1)s

for(l=0; l(n; l++)) and = max(ans, mat[e][m-1])
   der (intien-1; ix 0; i=) (i) (i) (i)
          for (int )= res-1; ] 120, 45-73 [] 130
              y (n i==(n-1) 2) j==(m-1)
                   dp[i][j] == mat[2][j];
               else if (i== (n-0) (0=> [n]qb) ; [];
               else of (j== (m-1))
                    ap [1] = mat [1] [];
               ولهو
                  de [i][]= 1+ min de [i][]+1], min de [i+1]
             ans= max (ans, op [t][]);
     return ans;
```

```
# Minimum Path Sum son of a say y started son top 1
 Movement to solve from bottom to top.
                     min (dp [i+1] [j], > down de [i] [j+1] > wight
                Lord Lord Letter ( litt " gut, int i, int file
  int dp[n][m];
  for (int i=n-1; i>=0; i--) of
      for (int j=m-1; j>=0; j--) }
          J(2==m-D) 42 j==(m-D)
             طه [دً] [ع عسد [دً] [ع];
         else y (i== (n-1)) 11 Movement night (dast Row)
              dp[i][] = dp[i+1] + auc[i][]);
         else y (j==(on-D) 11 Movement down (dast col)
       ap [2] [3] = ap [a) [3) + aux[2] [3]
        else
          dp[e][] = min (dp[e+1)[], dp[e)[]+1])
                        + auctosfoj;
   vieture de Coscos; las Juin + [3] rais - slov in
               suction de [[][] = max (vall, vald);
            long lang max Houst ( but awel), but on)
                       number (dp, -1, & speed (dp));
                      section solve (and, O, 21-1);
```

```
#) Optimal Storategy for a game
                 got at motival marely subsect training
  Cadle
  "int aption I [100];
  long long solve (int + aue, int i, int j) of
       y(is j)
                           المعروس الم ١٠٠١ الله ١٠٠١ الله ١٠٠٠ الم
         return 0;
      y (de [i) [j] [=-1)
                        matum apring; (1-10=3)
      11 of I take i; opponent chooses fever (c+1-)
       Opponent true to take val that makes opt for me mi
 opponent takes (i+1) > I have (i+2) -> j
       of opponent takes (i) -> I have (2+1) -> (i-1)
      int val = auctij + min ( solve (aux, î+2, j));
      11 4 d take j', apponent chouses form (c, j-)
       of opponent takes i; I choose fevora (i+1, j-1)
       opponent takes j-1; ? choose ferom (21/2)
       int vald = aue [ ] + min ( solve (aue, 2+1, 9-1))
       vietur dp[2][j] = max (vall, vald);
    long long max Amont (int awet), int only
     memset (dp, -1, size of (dp));
     setwer solve (aver, 0, n-1);
  3
```

```
#) Maximum Pengit from sales of wine
 Each yee you can see only the last and first wine.
 On yth year; ith wine will be y+P2
  Code
 int max Purgit (int wine [], int s, int e, int yei) of
     of (8 = 1 pt) do [8] [e] ]=-1)
         return you * dp[s][e);
     y (8== e)
         return yer + perie [8];
                                 2 3 .0 3
     int left = (peuco[s] + yes) + max Bugit (puice, s+1, e, yer+);
     int eight = (pulce[e] +yer) + max Pusfit (puice, 8, e-1, y
     metun dp [s)[e] = max (left, mght);
  int wine (int puivo E), int n)
      dp[n][n];
     menset (dp, -1, 812eof (dp))
     return maxPerofit (puice, 0, n-1, 1);
```

```
# Maximum Sum Reitangle in a 22 mateux
   application of Kadane's algolithm in 20 - four.
          (j=2, j<R)
       Scrip (K=0; ) LACO ( Lever the) flow
  Cade
   int max dum Rectangle (int R, int C, V/V/int) M)
       int max = INT_MINT;
for Cint i=0; ix R: i++) & [2] wing = the
   forter en possible rectangles.
          for (int j= 03) x Rij ++) &= (0102) 10
              for(int k=0; kxC; k++)&
                 V[K) += M[j][K]; sing in
              max = max (max, kadane (V,C)
                       chun max/ infit (puice,
      wetwer max;
```

```
# Regular Expussion matching
                             @ Bor pc == 3c
                                    والع الم (م د حدد
    B for '*'
           18t Col:- loobe & up. (c-2).
               dlooked up comp. y p[i-2] == scall
            dp[2][j] = dp[2-2][j] 1 = ( ) | p[2-2] == )
            Code bool dp[n+i][m+i];
                             n -> pattern. lengthes;
                             m -> strung, length ();
 for(1=0; 12n+1; 1++)&
    for (int 1=0; jam+1; j++) of
        ŷ(i==0 22 j==0)
Op[i][j]=+eue;
                              Il first coll
       ely (i==0)
                             man turif !
           dp[e][j] = false;
       else y (j==0) &
                             Il first column.
           char fc = p[e-i];
          of (bc== " *")
              dp[=][]=dp[=2][];
             dp[i][j]= false;
       else
                                110 there.
         char pc=p[i-i];
         char &c = &[j-1];
         g (pc == 1 * 1) &
             op[ [] [ ] == op [ 2 2] [ ];
             char pest = op[i-2];
             y (pcsl = "· ' 11 pcsl = = sc)
                of [1] [] = of [1] [] | of [1] []-1];
        3
```

else y (pc == '.') dp [ ] = dp [ - ] [ - ]; else y (pc==&c) olp [1] [] = olp [1-1] []-1); else dp[i][j] = false; pattern. long HL); 1 96 [4-1] [w+1]? return ap[n][m]; Ex 8.4 1; E++) & \$ (+4) items i ;0 = 1 de [[][] = time; apter falles 1 (0==1) Bs chan be = P[c-1]; ("x"=='29) & of [62] = aple-31 [3]: (app = Col Col do)

Scanned with CamScanner

```
# Wild Card Pattern Matching to top.
                             (C+1) (P+D) Q PC == 30 D 12 D 13
                           3 of pc == '+'
                                                     last col -> dp[estjs=dp[e+1][j].
                                                     elle

→ de [ερτή = de [ε+ή τ΄) 11 de [ερτή+ή].
          Code bool dp[n+][m+]. n-> pattern. length();
           for (°= n A; i>= 0; i-) { m > stung. length();
               Jor (j=m; j=0; j--) & (j=1) plant com de [i] [j] = time; ll dast com de [i] [j] = time; ll dast Row de [i] das
                   Olse ? (j==m) { 11 das+ col. } (p[i] == )*') dp[i][j] = dp[i+D[j];
                                              else de CEDGO = false; original)
                                 else
                                                Char pc=p[e]; char sc=s,[];
                                               ¥ (PC== \*')
                                                                dp [ 2] [ ] = dp [ 2] [ ] | dp [ 2+ ) [ ];
                                            else y (depc==) 21)

de [2] [3] = de [2+1) [3+1);
                                              else & (pc== sc)
of [e) [] = dp [e+0[]+0;
                                                else
                                                            olptistis = false;
      return of [0] [0];
```

```
# Matiex Chain Mutiplication
SIP:- 240,20,30,10,30}
                            019:- 26000.
 Code
                              0(N3)
   int ofptioi)tioij;
   int salue (int + au, int 2, int j) of
      y (1)= )
      vietum 0;
     g(dp[e)[]=-) =-1 (]=(3)
         vetuen de [?][];
      int ma= INT_NAX; and (7)
      for (int k= 2#1; kxj; k++) & 1
          int tempAns = solve (aux, i, k) + solve (aux
                     + auc[e-1] * auc[k] * auc[j:
       : ( ) ( ) 1 po - ( ) ( ) our l'individuel
         if (tempAns x min) component!
             mr=tempAns;
      return, jolp [2] jum.; [3] q = 00 10)
  int MCNL (int # aux), int my
       memeet (dp, -1, 8920 of (dp));
       int = 1; 1= 0 = cpcope
       retwer solve (aux, c, j);
                    olse delicités : false;
```

```
# Dungean Grame
                         11, Him program opulary
 Code
  int n = dungeon . 2020c);
  int row = dugeon [0]. 8P2e (); introversitory with a giru
  ind of [red] [ma] () soit. Bill of shortens on
  Jor(ut i=n-1; i>=0; i-) gtormon : [m][n] po to
      for(it ]=m-1; }>=0; ]-->&
          y(i==(m-0))
              ap[es[i] = min(o, dugeon[is[i]);
           else y (2==(n-1))
             de[i][j] = min(0, ap[i][j+j]+ dung eon[i][j]);
           else y (j==(on-1))
              de [[][] = rain (0, de [[+]][]) + durg eart[][]);
             de [e][g] = min(o, max(de[e+1][o]) + dep[e][j]);
       3
                               :1 = (5)(0) to
    3
   uetwer aboldplos(0)) + 1;
                   for ( at 1=1: 1 m: 1+96
              G== C/1(,1 price) oporso) h
                       of [12] - 0;
     $ 0 20 20 40 + 120 - 2 49 = ches 249
                            reprine ap [0-1][01.1):
```

```
# Unique Paths - I (dect Code - 63)
                                             Dynamic Per
     Obstacles marked with '1';
                                      comale nespeut
   Cado
    int unique Paths (vector (vector (int)) & obstacle Grewer,
        int n= obstacle Grand. Size (); [ m] [ m]
        int m=obstacle Grand [0]. Size ();
                         memset (dp, 0, size of (dp));
         int of [n][m];
        111st Col. - of you encounter a obstacle
          We way to heach well below.
         for(int 1:0; ixn; i++)&= (70396
              y CobstacleCrewd [i)[0] == )
                  brushe;
1+11+11(3) qb ,0) rum = []
 : (C) (C) na gelse
                dp [i][0] = 1;
               op [0)[9] = 1;
        3
       for (int i=1; ixx; i++) & (0)(0) qb)(do
           for (int j=1; jx m; j++) of

y (obstacle (y sud [i) [j) == )

dp [i)[j) = 0;
                     dp [25] = dp [2-1)[3) + dp [2:7[3-1);
       entwer dp to-1) [m-1);
  2
```