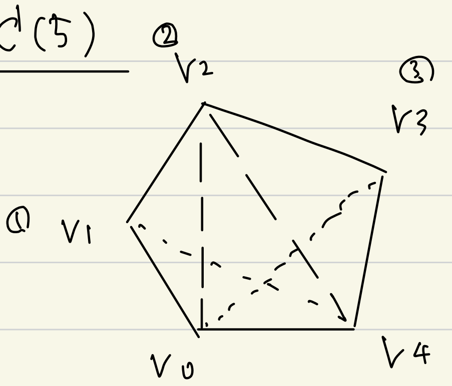
R1 $\boxed{1}$

(1)(2) $C(2)=1$, $C(3)=1$, $C(4)=2$ は既知とする。

$n \geqq 5$において、$C(n)$の値を考える

$\underline{C(5)}$



この時、v_0v_4を辺に含む3角形の3頂点のうち、v_0, v_4のいずれでもないものがどれであるかにおいて場合分けする。

① v_1の場合：残る分割は4角形 $v_1v_2v_3v_4 \to C(4)$

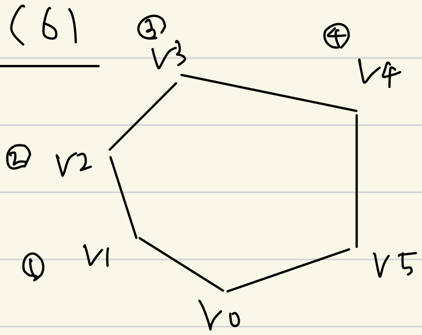
② v_2の場合：残る分割は3角形$v_0v_1v_2$と3角形$v_2v_3v_4$

$\to C(3)C(3)$

③ v_3の場合：残る分割は4角形 $v_0v_1v_2v_3 \to C(4)$

ゆえに $C(5) = C(4) + C(3)C(3) + C(4) = \underline{5}$

$\underline{C(6)}$



同様にv_0v_5を辺に含む3角形の3頂点のうちv_0, v_5に属さないものが、$v_1 \sim v_5$のどれであるかにおいて場合分けする。

①の場合：$C(5)$　③の場合：$C(4)C(3)$

②の場合：$C(3)C(4)$　④の場合：$C(5)$

ゆえに $C(6) = C(5) + C(3)C(4) + C(4)C(3) + C(5)$

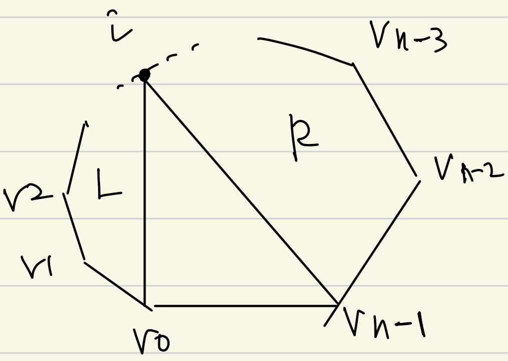
$= \underline{14}$

$\underline{C(7)}$ まったく同様に

$C(7) = C(6) + C(3)C(5) + C(4)C(4) + C(5)C(3) + C(6)$

$= \underline{42}$

$d(n)$ を $d(2) \dots d(n-1)$ を用いて漸化式として表現する。
これまでの考え方を応用する。



$v_0 v_{n-1}$ を含む3角形の3頂点のうち、$v_0 v_{n-1}$ のいずれでもないものが v_i ($1 \le i \le n-2$) のいずれかであるかに対して場合分ける。便宜上、2角形 $= \phi$ として定めれば、3角形 $v_0 v_{n-1} v_i$ により、もともとの多角形は左図の $i+1$ 角形領域 L と、$n-i$ 角形領域 R に分割される。つまり、残る範囲には $d(i+1)d(n-i)$ 通りの分け方がある。

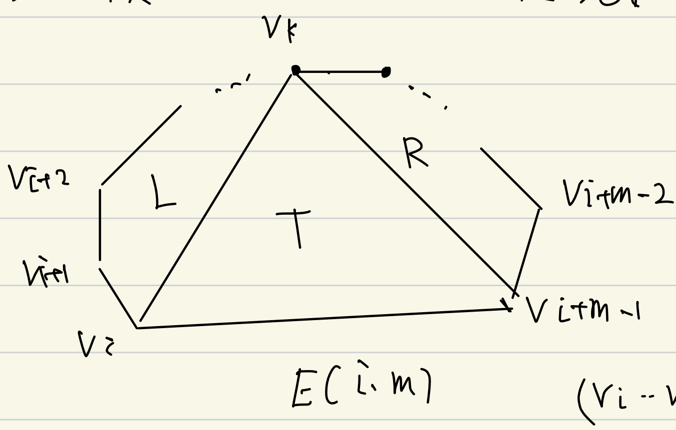
ゆえに $$d(n) = \sum_{i=1}^{n-2} d(i+1)\, d(n-i)$$

(3) (2)の漸化式に合うように擬似コードを完成させると以下のようになる。

```
d[2] = d[3] = 1
for i {4 ... n}
   d[i] = 0
   for j {0 ... i-3}
      d[i] = d[i] + d[j+2] d[n-1-j]  .. ①

return d[n]
```

(4) 以下便宜上 $v_{i+n} = v_i$ と定める。



$E(i, m)$

右図のように $E(i,m)$ を小問題にわけて考えるため $v_i, \dots v_{i+m-1}$ の m 個の頂点からなる多角形を左右に分割するような3角形 T: $v_i v_k v_{i+m-1}$ ($i < k < i+m-1$) の k ごとに考える。図より、3角形 T により、もともとの多角形は ($v_i \dots v_k$ からなる多角形 L) と、($v_k, \dots v_{i+m-1}$ からなる多角形 R) に分割され、それぞれの分割最小コストは明らかに独立である

つまりこの時
$E(i, m)$ = (L の分割コスト) + (R の分割コスト) + (三角形 T のコスト)

$\therefore\ E(i,m) = E(i,k-i+1) + E(k,i+m-k) + D(i,k) + D(k,i+m-1) + D(i,i+m-1)$
という関係式が成り立つ. ゆえに, $i<k<i+m-1$ に注意して

$$E(i,m) = \begin{cases} 0 & (m=2) \\ \min_{i<k<i+m-1} \{E(i,k-i+1)+E(k,i+m-k)+D(i,k)+D(k,i+m-1)+D(i,i+m-1)\} & (m>2) \end{cases}$$

(なお, 便宜上 $V_i = V_{i+n}$ と定めていることに注意)

(5) (4)の結果より $E(i,m)$ の値は $m'<m$ をみたすような任意の j についての $E(j,m')$ の計算結果から求められる。$E(i,2)=0$ に注意して $m=2,\cdots n$ の順で $E(i,m)$ を計算してゆく疑似コードを示すと以下のようになる.
なお, コード中の // (コメントアウト) にその補足を示す.

```
for m in {2 ... n}
    • if m = 2 : E(i.m) = 0  for all 0≤i<n ; continue;
    // E(i,m) を初期化
    E(i,m) = ∞  for all  0 ≤ i < n
    for i in {0, ... n-1}
    for k in {i+1, ... i+m-2}
    // (4)の漸化式より E(i,m) を計算
    E(i.m) = min{E(i,m), E(i,k-i+1) + E(k,i+m-k) + D(i,k) + D(k,i+m-1)
                 + D(i,i+m-1)}
// 求める最小コストは E(i,n) (iは自由) より, E(0,n) とする.
ans = E(0,n)
return ans
```

なお, 便宜上 $V_{i+n} = V_i$ と定めていることから $D(i,j) = D(i\%n, j\%n)$ に等しいことに注意する.
上記の疑似コードのアルゴリズムについて考える. まず m に関する $O(n)$ のループが発生し,

その中で i に関する $O(n)$ のループが呼ばれ、各 $E(i,m)$ を k に関する $O(m)$ のループで更新している。全てのループ内での処理は $O(1)$ での比較のみであるため、疑似コード全体の計算量は

$$\sum_{m=0}^{n}\sum_{i=0}^{n}\sum_{k=1}^{m} 1 = n \cdot \sum_{m=0}^{n} m \leqq n\sum_{m=0}^{n} n = O(n^3)$$

より、$O(n^3)$ であると計算できる。