*SOL OF WORD*

Gọi F[i, j] là chi phí được khi xóa hết đoạn i -> j.

Gọi L[i] là chi phí lớn nhất nhất khi xét đến vị trí i trong xâu S.

* L[i]=max { L[i – 1], L[j] + F[ j , i ] } (j < i).
* Kết quả là L[S.size()].

Công thức truy hồi tạo mảng F[i, j]

* ***Subtask 1, 2:*** F[i, j] = max{ F[i, k] + F[k+1, j], F[i+1, k-1] + F[k+1,j – 1] + w(i,j,k). (Chi phí xóa xâu[ijk]).
* ***Subtask 3, 4:*** G[i, j, k, h] là chi phíxóa xâu từ i -> j còn lại đúng h kí tự đầu tiên của kí tự Wk .
  + G[i, j, k, h] = G[i, j’, k, h – 1] + F[j’+1, j] (j’ < j).
  + G bổ sung cập nhật thường xuyên cho F.
  + Ngoài ra cần cập nhật mảng F thường xuyên :
    - F[i, j] = max(F[i, j], F[i, k] + F[k+1, j]).
  + Khi W(k).size() == h thì cập nhật
    - F[i, j] = max( F[i,j], G[i, j, k, h] + p[k]).
    - G[i, j, k, h] + = p[k].
  + Chuẩn bị thêm mảng pos[i, c] là vị trí lớn nhất không vượt quá i có kí tự c.
* ***Subtask 5:*** Vận dụng ý tưởng của sub 3,4.
  + Gọi G[i, j, node] là chi phí xóa xâu từ i -> j mà cặp (k, h) lúc này được biểu diễn bằng node trên câu TRIE.
  + G[i, j, node] = G[i, k, node’] + F[k+1,j-1].