* Thuật toán cập nhật đỉnh con lên đỉnh cha trong cây.
* Ta gọi in[u], out[u] lần lượt là thứ tự vào và thứ tự ra của đỉnh u trong quá trình dfs từ gốc,
* Ta có thể in[u], out[u] qua thuật toán dfs :  
  void dfs(int u) {  
   in[u] = ++dem; a[dem] = u;  
   for là con u : dfs(v);  
   out[u] = ++dem; a[dem] = u;  
  }
* Ta nhận thấy trên mảng a, vị trí các con của u nằm gọn trong in[u] và out[u].
* Nhận xét trên là cơ sở để ta sử dụng IT thực hiện các thao tác cập nhật từ đỉnh con lên các đỉnh cha của nó trong cây IT. Khi thực hiện 1 thao tác cập nhật nút cùng các cha của u ta thực hiện thao tác update(in[u]), khi thực hiện thao tác lấy giá trị 1 nút ta get(in[u], out[u]).
* Ta có thể sử dụng thuật toán trên kết hợp chặt nhị phân để làm sub2 bài salary (O(). Để qua được sub 3 cần cải tiến hàm get kết hợp với LCA để loại bỏ thao tác chặt nhị phân giảm đpt xuống O().
* Sử dụng cấu trúc dữ liệu đơn giản như stack, deque tìm min max trên đoạn tịnh tiến … hay các thuật toán tìm vị trí trái nhất, phải nhất… tuy code đơn giản nhưng chạy nhanh hơn rất hiều so với IT (O(n) thay vì O(nlog(n))), và giải quyết được rất nhiều bài toán.
* Với mỗi loại bài tập cần cho học sinh làm 4 đến 5 bài để đánh giá dc loại bài đó (về độ phức tạp tính toán, thời gian thực hiện, code dài hay ngắn, có dễ sinh lỗi hay rơi vào các trường hợp xấu không….)