

$$\textcircled{1} \quad \langle \underline{1}, \underline{0}, \underline{0}, \underline{1}, \underline{0}, \underline{1}, \underline{0}, \underline{1} \rangle, \langle \underline{0}, \underline{1}, \underline{0}, \underline{1}, \underline{1}, \underline{0}, \underline{1}, \underline{1}, \underline{0} \rangle$$

$$\rightarrow \langle 1, 0, 0, 1, 1, 0 \rangle$$

$\textcircled{2}$ ابتدا دنباله اعداد را Sort می کنیم $(O(n \lg n))$ ؛ سپس LCS دنباله‌ی مرتب شده و دنباله‌ی اولیه را پیدا می کنیم $(O(n^2))$. $\leftarrow O(n^2)$

$\textcircled{3}$ OPTIMAL - BST : تغییر در خط 10 الگوریتم $\text{for } r = r[i, j-1] \text{ to } [i+1, j]$ با در نظر گرفتن اثبات Knuth، می توان حدود حلقه را طوری تغییر داد که اشتباه نداشته باشند.

$\textcircled{4}$ اگر داد که به صورت p (عدد دلخواه) $d \leq 1$ باشد، الگوریتم greedy همیشه جواب optimal می دهد؛
فرض کنید قرار است مجموع n سکه سود می دانیم: $n = pq + r$ ($q \geq 0$) که برابر است با q که به ارزش p و r که به ارزش 1 .
الگوریتم greedy ابتدا تعدادی سکه به ارزش p و سپس بقیه سکه‌ها را به ارزش 1 انتخاب می کند $= pq + r$ پس optimal است.