الف) $T(n) = T(n-1) + n$

حالت پایه $T(1)$ است پس می‌توانیم رابطه را بسط دهیم تا به این حالت برسیم:

$$T(n) = T(n-1) + n$$
$$= T(n-2) + n-1 + n$$
$$= T(n-3) + n-2 + n-1 + n$$

ادامه دادن این الگو تا جایی که به حالت پایه برسیم:

$$T(n) = T(1) + 2 + 3 + \dots + n$$

با فرض اینکه $T(1)=1$ این معادله برابر مجموع اعداد ۱ تا n میشود:

$$T(n) = 1 + 2 + 3 + \dots + n = n(n+1)/2$$

پس:

$$T(n) \sim O(n^2)$$

درخت بازگشتی: درختی با ریشه با سایز n داریم که هر راس دارای یک فرزند با سایز یکی کمتر است است. این درخت دارای عمق n است. و هزینه هر مرحله از این درخت از $O(n)$ است و چون دارای n مرحله است، پس از O $(n*n)$ است.

ب) $T(n) = 2T(n/2) + n$

درخت بازگشتی:

ارتفاع ۰ ← دارای هزینه n

ارتفاع ۱ ← دو راس با سایز $n/2$ پس هزینه این ارتفاع $2*(n/2) = n$ است.

ارتفاع ۲ ← چهار راس با سایز $n/4$ پس هزینه این ارتفاع $4*(n/4) = n$ است.

به طور کلی هر ارتفاع دارای هزینه n است و این درخت تا جایی ادامه پیدا میکند که سایز مسأله به ۱ برسد. پس این درخت دارای عمق $\log_2(n)$ است. زیرا در هر ارتفاع ما n را تقسیم بر ۲ می‌کنیم و لگاریتم n بر پایه ۲ نشان دهنده این است که n را چند بار می‌توانیم به ۲ تقسیم کرد.

پس هر مرحله دارای هزینه n است و درخت دارای عمق $\log_2(n)$ است در نتیجه کل هزینه برابر است با:

$$T(n) \sim O(n*\log_2(n))$$

امیررضا محمدی یگانه