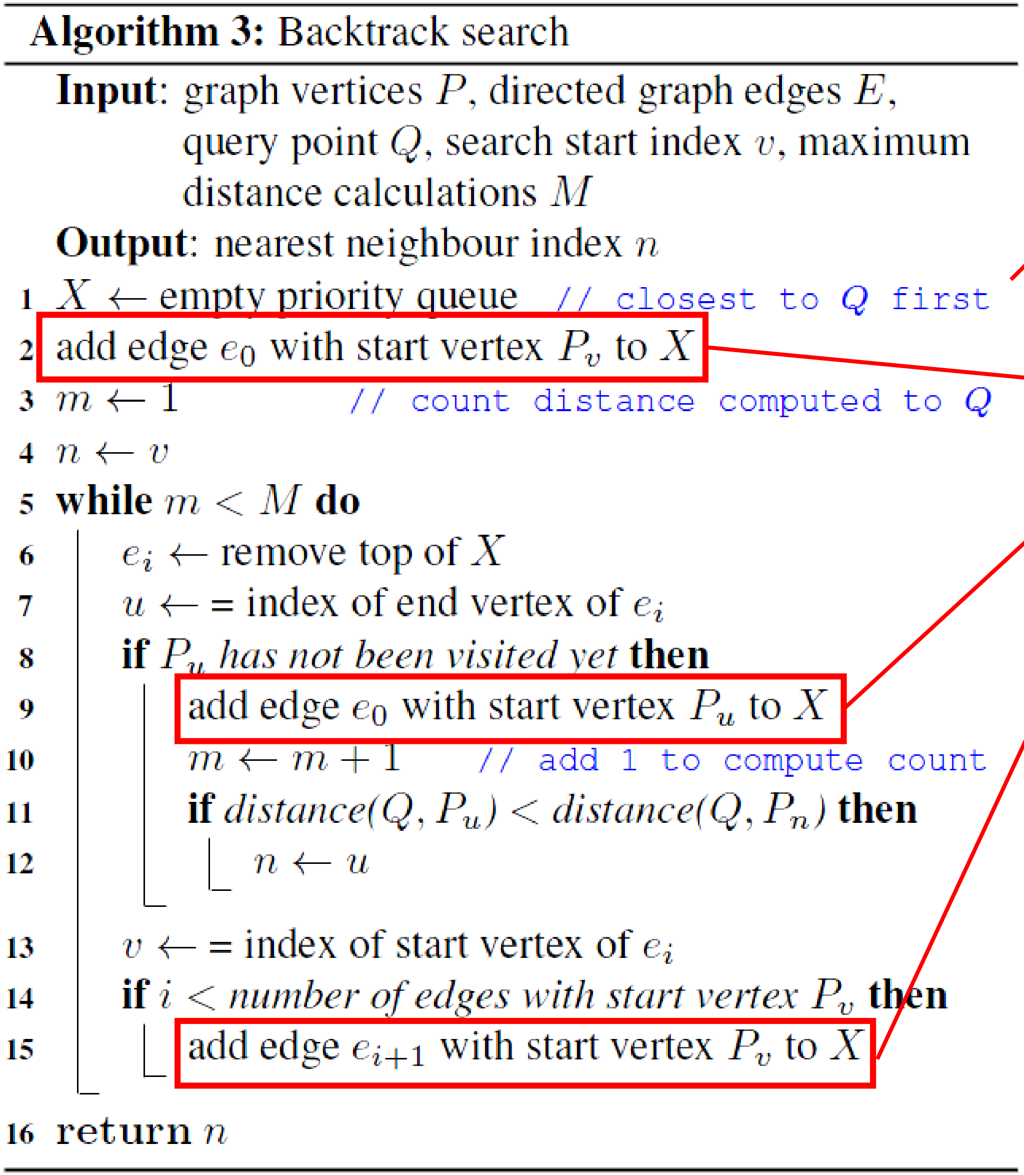

Algorithm 3: Backtrack search

```
Input: graph vertices $P$, directed graph edges $E$,
       query point $Q$, search start index $v$, maximum
       distance calculations $M$
Output: nearest neighbour index $n$
$X \leftarrow$ empty priority queue   // closest to $Q$ first
add edge $e_0$ with start vertex $P_v$ to $X$
$m \leftarrow 1$   // count distance computed to $Q$
$n \leftarrow v$
while $m < M$ do
    $e_i \leftarrow$ remove top of $X$
    $u \leftarrow$ = index of end vertex of $e_i$
    if $P_u$ has not been visited yet then
        add edge $e_0$ with start vertex $P_u$ to $X$
        $m \leftarrow m + 1$   // add 1 to compute count
        if distance($Q, P_u$) $<$ distance($Q, P_n$) then
            $n \leftarrow u$
    $v \leftarrow$ = index of start vertex of $e_i$
    if $i <$ number of edges with start vertex $P_v$ then
        add edge $e_{i+1}$ with start vertex $P_v$ to $X$
return $n$
```

根据start vertex到query的距离

每个has not been visited 的节点在初次处理时，先根据OSH进行排序，此后按照OSH的顺序加入