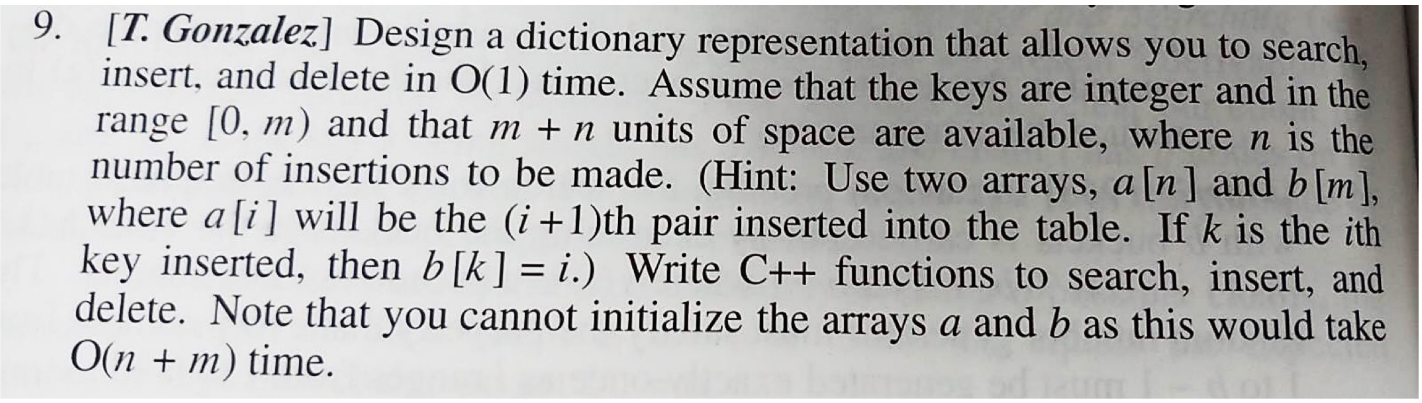
6. [*T. Gonzalez*] Let $s = \{s_1, s_2, s_3, ..., s_n\}$ and t = $\{t_1, t_2, t_3, ..., t_r\}$ be two sets. Assume $1 \leq s_i \leq m$, $1 \leq i \leq n$, and $1 \leq t_i \leq m$, $1 \leq I \leq r$. Using the idea of Exercise 9, write a function to determine if s ⊆ t. Your function should work in O($r + n$) time. Since $s \equiv t$ iff s ⊆ t and t ⊆ s, one can determine in linear time whether two sets are the same. How much space is needed by your function?



9. [*T. Gonzalez*] Design a dictionary representation that allows you to search, insert, and delete in O(1) time. Assume that the keys are integer and in the range $[0, m)$ and that $m + n$ units of space are available, where n is the number of insertions to be made. (Hint: Use two arrays, $a[n]$ and $b[m]$, where $a[i]$ will be the $(i+1)$th pair inserted into the table. If k is the ith key inserted, then $b[k] = i$.) Write C++ functions to search, insert, and delete. Note that you cannot initialize the arrays a and b as this would take O($n + m$) time.

```
int a[n], b[r]

 for i=0~n
    a[s[i]] = i
 for i=0~r
    b[i] = r[t]

=> check if s[i]=k is a subset of t
    if b[a[s[i]]] == -1
        return false
    else
        return true.

 (unordered_map)
```