

Csa 101 5-7-26

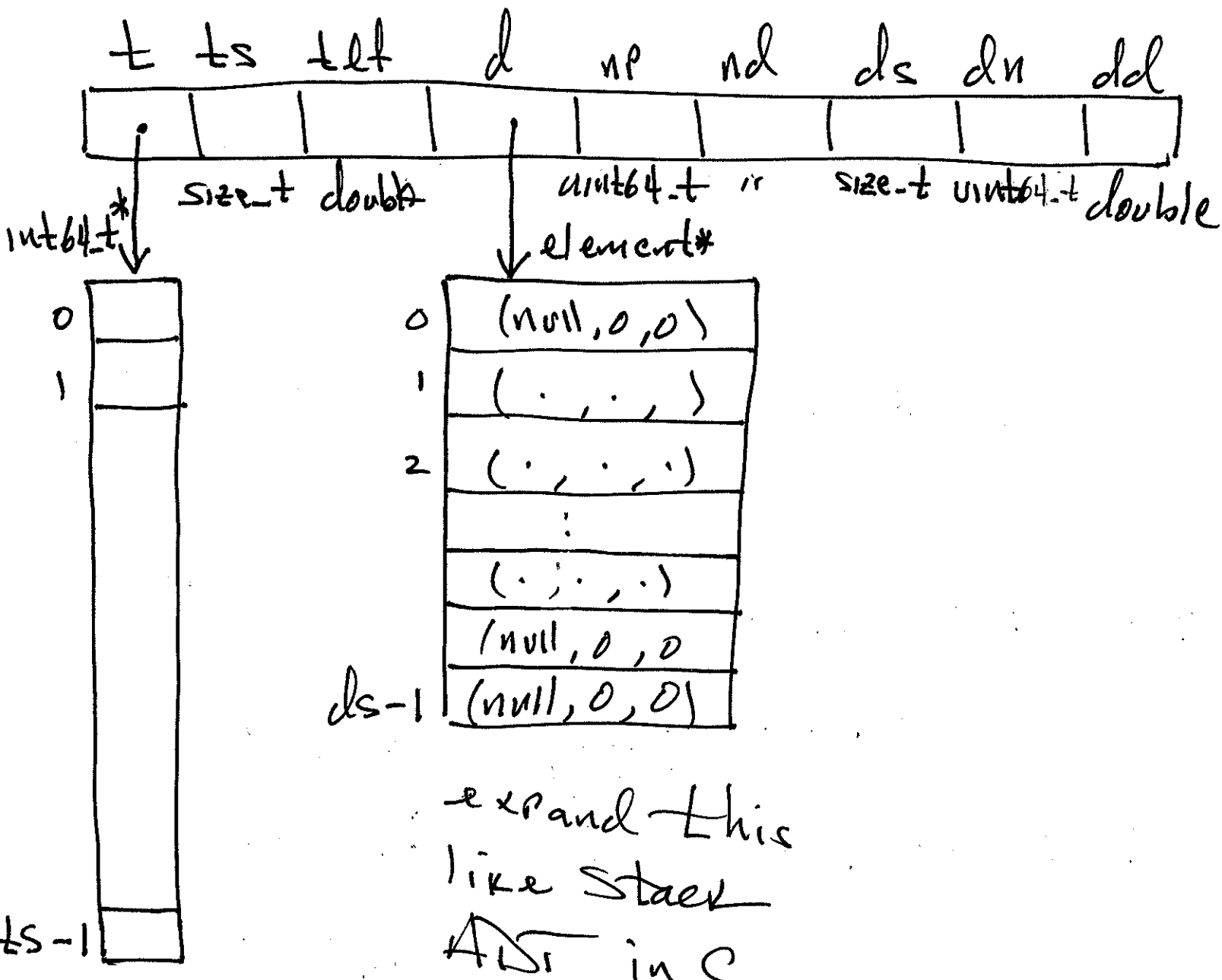
L

pas: ext. 2 days → Wed. 10 PM

Data array stores Example

```
typedef struct {  
    keyType key;           m.h: const char*  
    valType val;          m.h: int  
    codeType code;        m.c: uint64_t  
} element;
```

Dictionary Object



expand this like Queue ADT in C

compacting data

data

0	null, 0, 0
1	., a, .
2	., b, .
3	◇, c, .
4	◇, d, .
5	., e, .
6	null, 0, 0
7	null, 0, 0
8	null, 0, 0

← next

data

0	n, 0, 0
1	., a, .
2	., b, .
3	., e, .
4	n, 0, 0
5	n, 0, 0
6	n, 0, 0
7	n, 0, 0
8	n, 0, 0

← next

Helper Functions

• required

hash()] do not alter
probe()

• optional

	Build order
findSlot()	1
insertIndex()	2
expandTable()	3
expandData()	4
compactData()	5

ADT OPS: see Project PDF

setValue(D, k, v)

- compute hash code for key k
- find table slot where it exists or to be placed
- find current idx in table slot

if key exists, overwrite existing val
also // key k does not exist

do insertion:

- expand data if necessary
- Put ($k, v, code$) in data
- expand table if necessary
- compact data if necessary