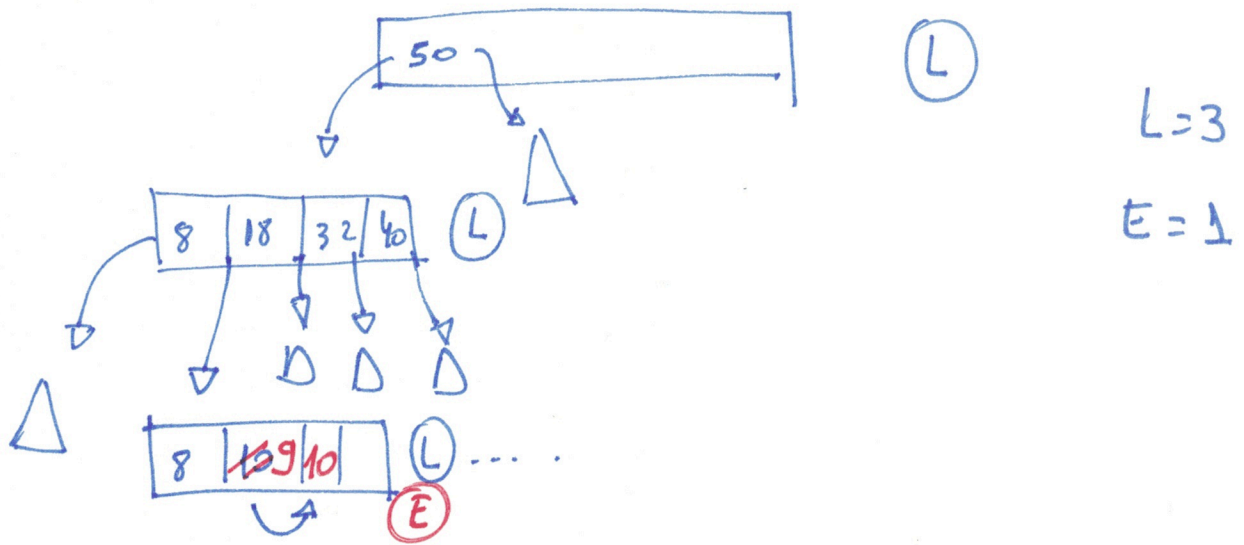
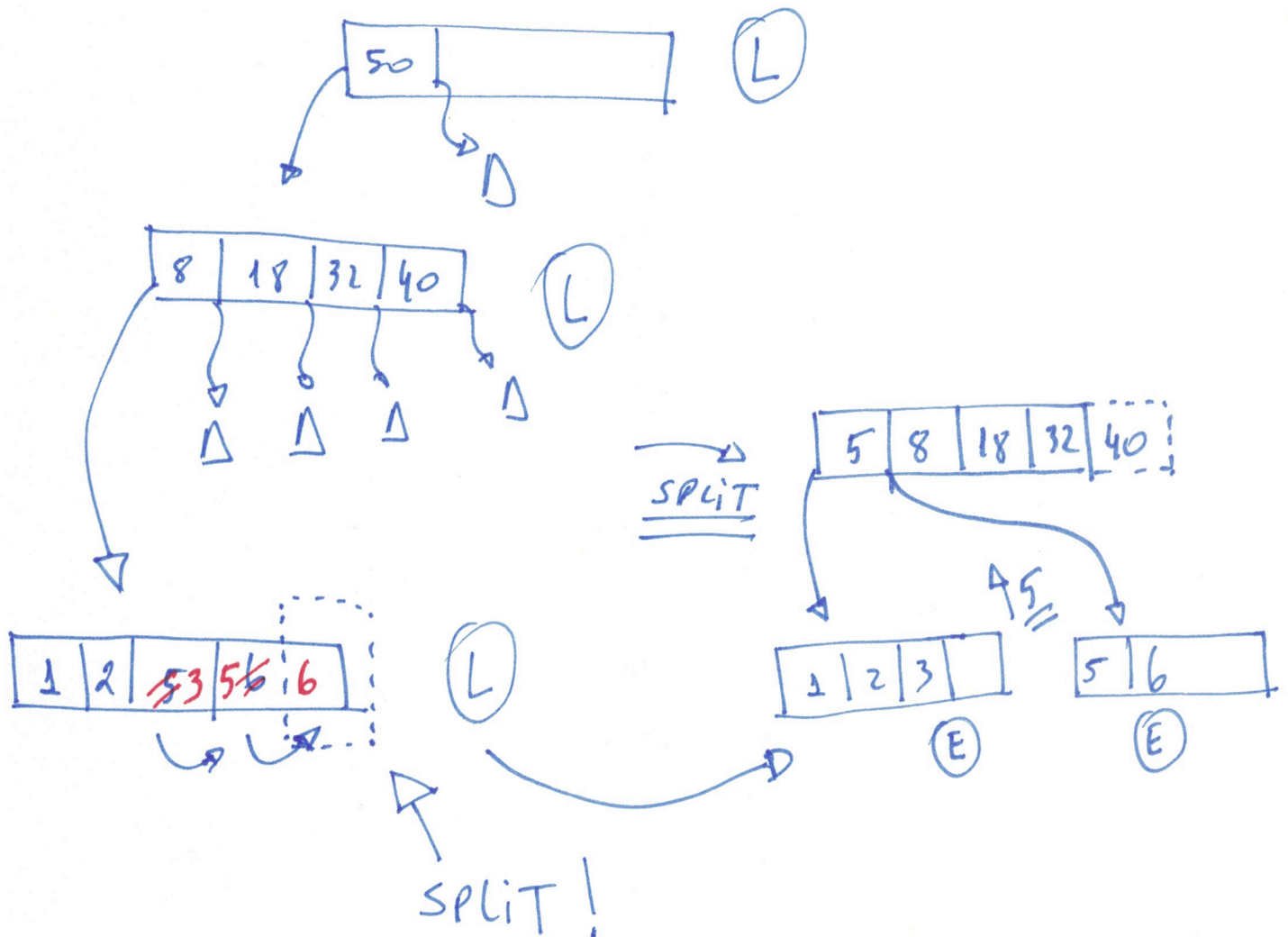


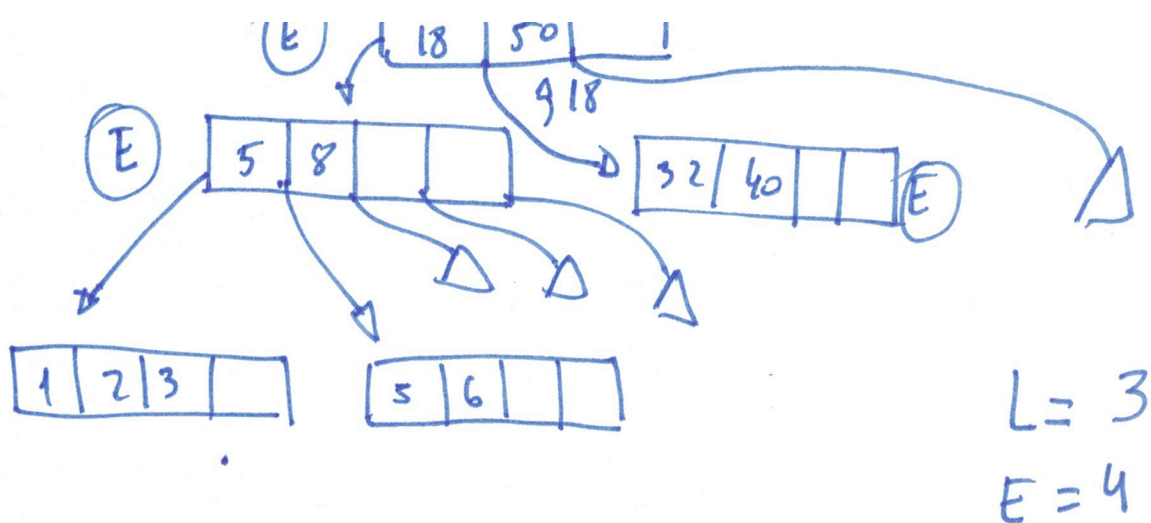
Questão 1:

(a) Inserindo 9.



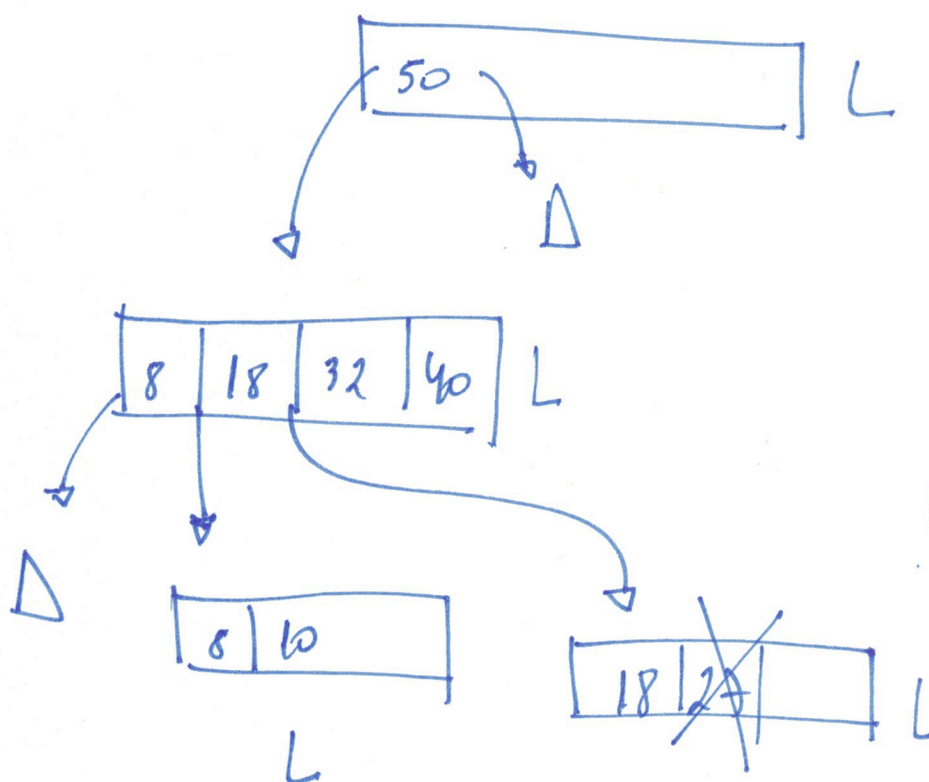
(b) Inserindo 3



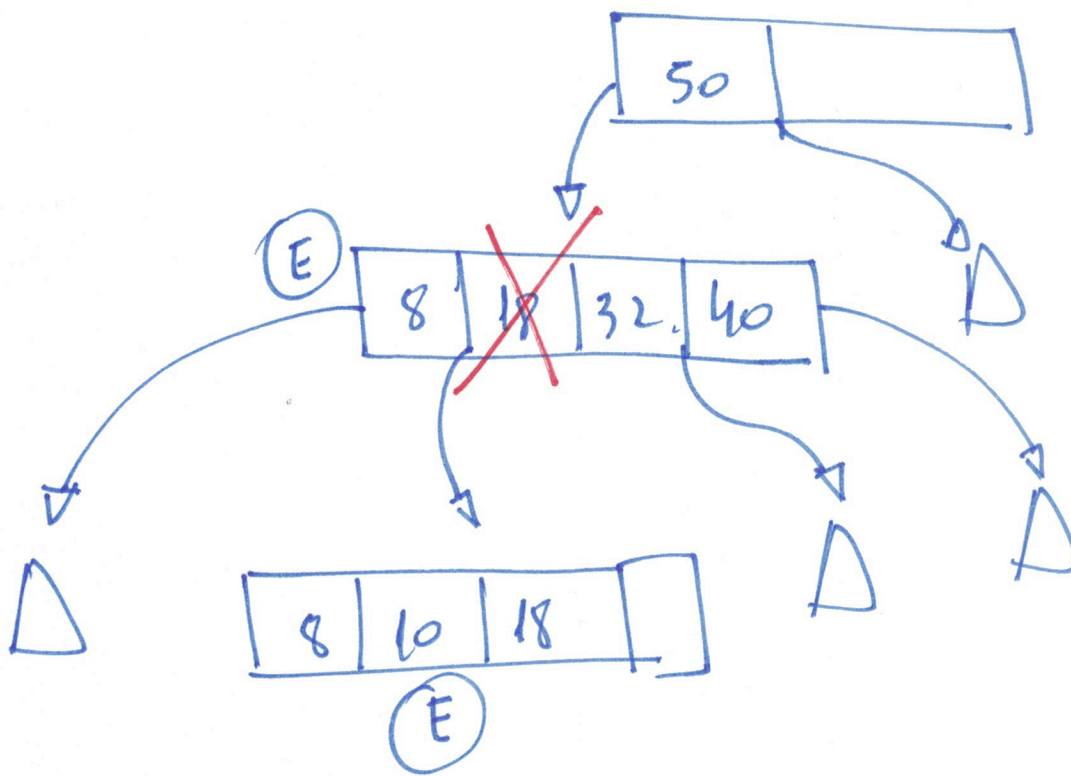


OBS: NA CORREÇÃO NÃO FOI RIGOROSO NO N° DE LEITURAS OU ESCRITAS.

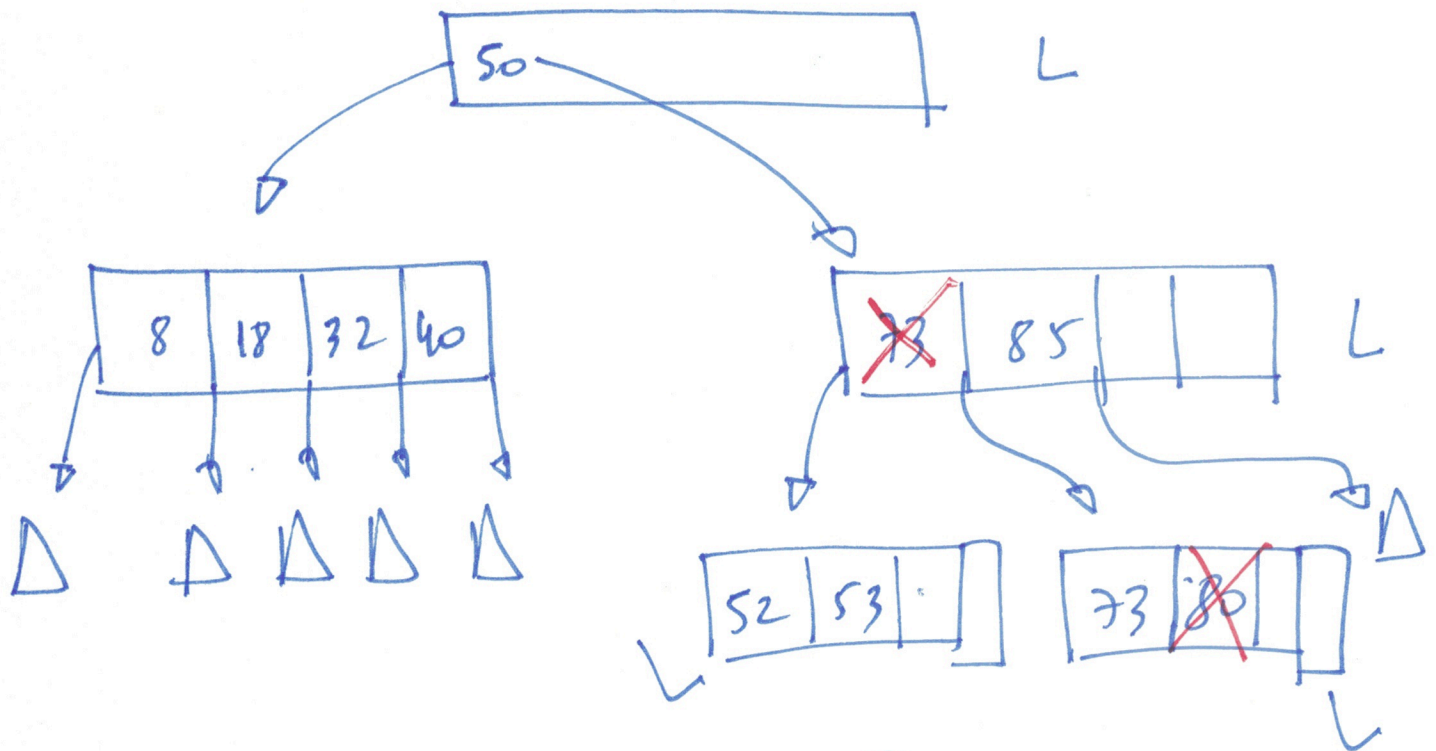
(c) REMOÇÃO DO 27



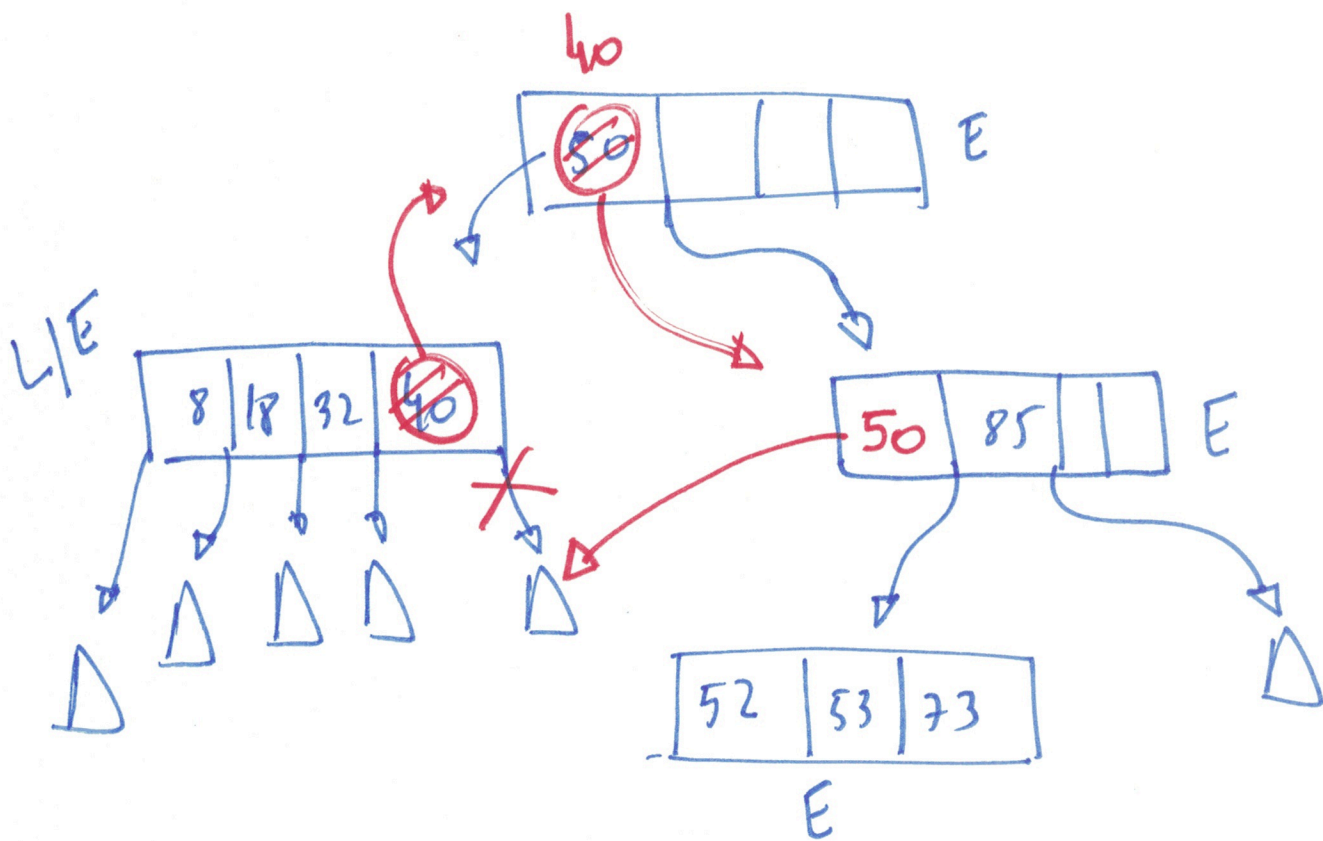
① Remove 27.  
 UNDERFLOW!  
 CONCATENAR  
 PAGINAS.



(d) remoção 80



① Remove 80  
underflow!  
CONCATENA.



$$L = 5$$

$$E = 4$$

Q2

B+

N chaves

N+1 ponteiros.

$$10N + 8(N+1) \leq 8192$$

$$18N \leq 8192 - 8$$

$$N \leq \frac{8184}{18}$$

$$N \leq 454.$$

(a) grau mínimo  $\approx \frac{454}{2} \approx 227$

aqui dei cento p/3 ou 4 //

b) - Nº

Paginas nas  
5 chas: (Aprox)

$$\frac{10^9}{2 \times 10^2} = \frac{10^7}{2}$$

$$\approx 5 \times 10^6 \text{ paginas.}$$

$$\log_{200} 5 \times 10^6$$

$$\frac{\log 5 \times 10^6}{\log 200} \approx 3$$



### Questão 3

```
indexName = "data/rep-hash.dat"
```

```
indexFormat = "8sLL"
```

```
indexStruct = struct.Struct("indexFormat")
```

```
fi = open(indexName, "rb")
```

```
fo = open("saida.dat", "wb")
```

```
linha = fi.read(indexStruct.size)
```

```
while (len(linha)):
```

```
    r = indexStruct.unpack(linha)
```

```
    if (r[0][0] != b'\0'): ← TESTA se  
                           posição VAZIA.
```

```
        fo.write(indexStruct linha)
```

```
    linha = fi.read(indexStruct.size)
```

```
fi.close()
```

```
fo.close()
```

b) No hash, As chaves estão espalhadas,  
portanto a ordem não pode ser determinada.

(c) Recriar a Tabela, reindexando todo o Arquivo.

Q4

pos	chave	prox
0	44	
1		
2	35	
3	47	
4	37	
5	16	
6	72	11
7		
8		
9	20	
10	21	

Colisão  
 { 11  
 12  
 13

(\*) Repita o processo até que  $\frac{M}{N \cdot K}$  se limite a uma única página.

Q5

Seja  $N$  o nº de chaves por página. Comece dividindo as chaves em  $\frac{M}{N}$  páginas, onde  $M$  é a quantidade de chaves. Pegue a menor chave de cada folha e agrupe-as em blocos de tamanho  $N$ , totalizando  $\frac{M}{N \cdot 2}$  blocos (\*)