SWAP

FUNZIONE AUSILIARE, MI SERVE PER SCAMBIARE DUE ELEMENTI:

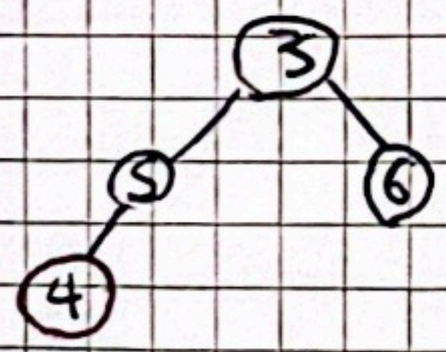
```
VOID SWAP(PRIORITYQUEUE:: PRIORITYQUEUE &PQ, INT POS1, POS2) {
    PRIORITYQUEUE:: ELEM AUX = PQ.DATA[POS1];
    PQ.DATA[POS1] = PQ.DATA[POS2];
    PQ.DATA[POS2] = AUX;
}
```

MOVE UP

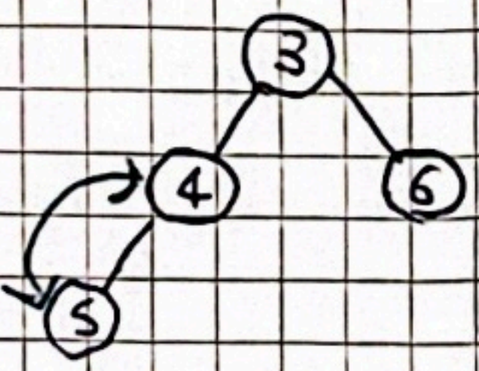
FUNZIONE AUSILIARE, MI SERVE PER RIPRISTINARE L'ORDINE DELL'HEAP, VIENE USATO NELL'INSERIMENTO PER PORTARE RIPRISTINARE L'ORDINE.

ESEMPIO:

INSERT(4);



MOVE UP



IL 4 E' PIU' PICCOLO DI 5,

CHIAMIAMO MOVE UP

DOVE L'HEAP E' SISTEMATO

CODICE:

```
VOID MOVE UP(PRIORITYQUEUE:: PRIORITYQUEUE &PQ, INT INDEX) {
    WHILE(INDEX > 0) { METTO > 0 PER EVITARE IL CASO CODA VUOTA
        INT PARENT INDEX = (INDEX-1) / 2   // QUESTA OPERAZIONE SERVE PER TROVARE
                                           INDICE DEL GENITORE
        IF(PQ.DATA[INDEX].TIMESTAMP >= PQ.DATA[PARENTINDEX].TIMESTAMP) {
            BREAK; }
        SWAP(PQ, INDEX, PARENTINDEX);
        INDEX = PARENT INDEX
```

INSERT

SE LA DIMENSIONE DI HEAP RAGGIUNGE HEAP.SIZE (SIZE == MAXSIZE), ANCHE DOPO AVER INGRANDITO LA CODA E RICHIAMATA MOVEUP:

```
PQ.DATA[PQ.SIZE] = ELEM
MOVEUP(PQ, PQ.SIZE);
++PQ.SIZE;
RETURN TRUE;
```

FINDMIN

IL SUO COMPITO È QUELLO DI RESTITUIRE L'ELEMENTO MINIMO (OVVERO LA RADICE) E NESSUN'ALTRO.

```
IF(ISEMPTY(PQ)){
        RETURN FALSE;
}
RES = PQ.DATA[0];
RETURN TRUE;
```

MOVEDOWN

FUNZIONE AUSILIARE UTILE PER LA DELETEMIN, IL SUO COMPITO È QUELLO DI RIPRISTINARE L'ORDINE DOPO L'ELIMINAZIONE.

PASSAGGI DETTAGLIATI:

1) CALCOLO DEGLI INDICI DEI FIGLI:

```
INT LEFTCHILD = 2 * INDEX + 1
INT RIGHTCHILD = 2 * INDEX + 2
```

QUESTI CALCOLI DETERMINANO GLI INDICI DEI FIGLI SINISTRO E DESTRO

2) INIZIALIZZAZIONE DEL PIU' PICCOLO

```
INT SMALLEST = INDEX;
```

ASSUME CHE L'ELEMENTO CORRENTE (ALL'INDICE INDEX) SIA IL PIU' PICCOLO.

3) CONFRONTO CON IL FIGLIO SINISTRO

SE IL FIGLIO SINISTRO ESISTE (LEFTCHILD < PQ.SIZE) ED È PIÙ PICCOLO

DELL'ELEMENTO CORRENTE, ALLORA SMALLEST PER ESSERE L'INDICE DEL FIGLIO SINISTRO

L'ELEMENTO CORRENTE SARÀ QUELLO ALL'INDICE INDEX (PASSATO COME

ARGOMENTO DELLA FUNZIONE).

4) CONFRONTO CON FIGLIO DESTRO

```
IF(RIGHTCHILD < PQ.SIZE AND PQ.DATA[RIGHTCHILD].TIMESTAMP < PQ.DATA[SMALLEST].TIME)
    SMALLEST = RIGHTCHILD
}
```

SE IL FIGLIO DESTRO ESISTE (RIGHTCHILD < PQ.SIZE) ED È PIÙ PICCOLO

DELL'ELEMENTO CORRENTE) ALLORA SMALLEST

DELEMIN

```
PQ.DATA[0] = PQ.DATA[PQ.SIZE - 1];
PQ.SIZE --;
MOVEDOWN(PQ, 0)
RETURN TRUE;
```

5) SCAMBIO E RICORSIONE

```
IF(SMALLEST != INDEX) {
    SWAP(PQ, INDEX, SMALLEST);
    MOVEDOWN(PQ, SMALLEST)
}
```

SE SMALLEST NON È PIÙ UGUALE A INDEX SIGNIFICA CHE UNO

DEI FIGLI È PIÙ PICCOLO DELL'ELEMENTO CORRENTE.

SCAMBIA L'ELEMENTO CORRENTE CON L'ELEMENTO PIÙ PICCOLO E CHIAMA

RICORSIVAMENTE MOVEDOWN SULL'INDICE DEL PIÙ PICCOLO PER

CONTINUARE IL PROCESSO.