1. Faça uma implementação do SelectionSort para listas ligadas.

```
p_no selection_sort(p_no lista) {
  p_no t, max, ant, ordenado = NULL;
  while(lista != NULL) {
    /*encontra o maximo*/
    max = lista;
    ant = NULL;
    for (t = lista; t->prox != NULL; t = t->prox) {
      if (t->prox->dado > max->dado) {
        max = t->prox;
        ant = t;
      }
    }
    /*tira da lista*/
    if(ant != NULL)
      ant->prox = max->prox;
    else
      lista = max->prox;
    /*insere no comeco da lista ordenada*/
    max->prox = ordenado;
    ordenado = max;
  }
  return ordenado;
}
```

2. Mostre um esquema para tornar qualquer algoritmo em um algoritmo estável. Quanto espaço e tempo adicional é necessário para o seu esquema?

Discussão em sala.

3. Crie uma versão iterativa de `desce_no_heap`

```
void desce_no_heap_iterativo(FilaP *fp, int k) {
  int filho = FILHO_ESQUERDO(k);
  while (filho < fp->n) {
    if (filho < fp->n - 1 && fp->v[filho].chave < fp->v[filho+1].chave)
      filho++;
    if (fp->v[k].chave < fp->v[filho].chave) {
      troca(&fp->v[k], &fp->v[filho]);
      k = filho;
      filho = FILHO_ESQUERDO(k);
    } else
      break;
  }
}
```

4. Em `desce_no_heap` trocamos um elemento com um de seus filhos e depois com um de seus netos, e assim por diante. Modifique as versões iterativas das duas funções para diminuir o número de atribuições (como feito no InsertionSort).

```
void desce_no_heap(FilaP *fp, int k) {
  Item t = fp->v[k];
  int filho = FILHO_ESQUERDO(k);
  while (filho < fp->n) {
    if (filho < fp->n - 1 && fp->v[filho].chave < fp->v[filho+1].chave)
      filho++;
    if (t.chave < fp->v[filho].chave) {
      fp->v[k] = fp->v[filho];
      k = filho;
      filho = FILHO_ESQUERDO(k);
    } else
      break;
  }
  fp->v[k] = t;
}
```

5. Mostre que o heapsort não é estável.

Basta simular o heapsort para $0, 1, 1, 2$**.**

6. Mostre que o quicksort não é estável.

Basta simular o quicksort para $1, 1, 2, 0$**.**

7. Argumente que o mergesort é estável.

Discussão em sala de aula

8. Suponha que você tenha um heap de máximo com todos os elementos distintos e com n elementos. Quais são as posições possíveis que o elemento mínimo pode ocupar nesse vetor?

Discussão em sala de aula