6.2 (3) There are $9 - 1 = 8$ passes.
After P=1: 1 3 4 1 5 9 2 6 5
P=2: 1 1 4 3 5 9 2 6 5
P=3: 1 1 2 3 5 9 4 6 5
P=4: 1 1 2 3 5 9 4 6 5
P=5: 1 1 2 3 4 9 5 6 5
P=6: 1 1 2 3 4 5 9 6 5
P=7: 1 1 2 3 4 5 5 6 9
P=8: 1 1 2 3 4 5 5 6 9

6.3
7-sort: 2 1 7 6 5 4 8 9 8
3-sort: 2 1 4 3 5 7 6 9 8
1-sort: 1 2 3 4 5 6 7 8 9

6.4
Because for merge sort, it has to do the same many number of comparisons no matter the input sequence, the input sequence does not matter too much. Let $T(n)$ denote the time needed to mergesort $n$ inputs.

Then, $T(n) = 2T\left(\frac{n}{2}\right) + n$

$T(1) = 1$

$\Rightarrow T(n) = n \log(n) + n$

The running time for sorted input, reverse-ordered input and random input is $n \log(n) + n$ when using merge sort.

6.6
(2) P=1, pivot = median (3, 5, 9) = 5
After P=1, it becomes 3 1 4 1 3 5 3; 5 9 5 6.
\[ T(N) = 2T\left(\frac{N}{3}\right) + cN \]

\[ T(N) = cN \log N + N = O(N \log N). \]

For random input.

Let \( S_1, S_2 \) be two subarrays.
Then size for \( S_i \) is equally likely, with a probability \( \frac{1}{n} \).

\[ T(N) = T(i) + T(N-i-1) + cN \]

\[ E(T(i)) = E(T(N-i-1)) = \frac{1}{N} \sum_{j=0}^{N-1} T(j) \]

\[ T(N) = \frac{2}{N} \sum_{j=0}^{N-1} T(j) + cN \]

\[ NT(N) = 2 \left( \sum_{j=0}^{N-1} T(j) \right) + cN^2 \]

\[ (N-1)T(N-1) = 2 \left( \sum_{j=0}^{N-2} T(j) \right) + c(N-1)^2 \]

Do the subtraction

\[ NT(N) - (N-1)T(N-1) = 2T(N-1) + 2cN - c \]

\[ \frac{T(N)}{N+1} = \frac{T(N-1)}{N} + \frac{2c}{N+1} \]

\[ T(N) = O(N \log N). \]

6.9

Stable: Bubble sort and Merge sort.

Unstable: Selection sort and Quick sort.
After $B$ is known $\Rightarrow$ After $G$ is known

<table>
<thead>
<tr>
<th>$V$ known</th>
<th>$dw$</th>
<th>$pv$</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>E</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>F</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>6</td>
</tr>
</tbody>
</table>

After $E$ is known $\Rightarrow$ After $F$ is known

<table>
<thead>
<tr>
<th>$V$ known</th>
<th>$dw$</th>
<th>$pv$</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td>E</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td>F</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>G</td>
<td>1</td>
<td>6</td>
</tr>
</tbody>
</table>

After $D$ is known

<table>
<thead>
<tr>
<th>$V$ known</th>
<th>$dw$</th>
<th>$pv$</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>D</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td>E</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>F</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>G</td>
<td>1</td>
<td>6</td>
</tr>
</tbody>
</table>

The distance from $A$ to other vertices:

$B$ 5 $C$ 9 $D$ 7 $E$ 8 $F$ 6

Maximum flow: $4 + 3 + 2 + 1 + 1 = 11$
Prim's Algorithm

After A is known

V known du pv
A 1 0 0
B 0 3 A
C 0 - 0
D 0 4 A
E 0 4 A
F 0 - 0
G 0 - 0
H 0 - 0
I 0 - 0
J 0 - 0

After B is known

V known du pv
A 1 0 0
B 1 3 A
C 0 4 A
D 0 2 B
E 0 2 B
F 0 - 0
G 0 - 0
H 0 - 0
I 0 - 0
J 0 - 0

After C is known

V known du pv
A 1 0 0
B 1 3 A
C 1 4 G
D 1 2 B
E 1 2 E
F 1 1 G
G 1 2 E
H 1 0 7 I
I 0 7 I
J 0 7 I

After D is known

V known du pv
A 1 0 0
B 1 3 A
C 1 4 G
D 1 2 B
E 1 2 E
F 1 1 G
G 1 2 E
H 1 0 7 I
I 0 7 I
J 0 7 I

After E is known

V known du pv
A 1 0 0
B 1 3 A
C 1 4 G
D 1 2 B
E 1 2 E
F 1 1 G
G 1 2 E
H 1 0 7 I
I 0 7 I
J 0 7 I

After F is known

V known du pv
A 1 0 0
B 1 3 A
C 1 4 G
D 1 2 B
E 1 2 E
F 1 1 G
G 1 2 E
H 1 0 7 I
I 0 7 I
J 0 7 I

After G is known

V known du pv
A 1 0 0
B 1 3 A
C 1 4 G
D 1 2 B
E 1 2 E
F 1 1 G
G 1 2 E
H 1 0 7 I
I 0 7 I
J 0 7 I

Final Configuration:

```
\begin{tikzpicture}
  \node (A) at (0,0) {A};
  \node (B) at (1,0) {B};
  \node (C) at (2,1) {C};
  \node (D) at (-1,0) {D};
  \node (E) at (0,-1) {E};
  \node (F) at (1,-1) {F};
  \node (G) at (2,0) {G};
  \node (H) at (-1,-2) {H};
  \node (I) at (0,-2) {I};
  \node (J) at (1,-2) {J};

  \draw (A) -- (B);
  \draw (B) -- (C);
  \draw (B) -- (D);
  \draw (B) -- (E);
  \draw (B) -- (F);
  \draw (B) -- (G);
  \draw (E) -- (H);
  \draw (F) -- (I);
  \draw (F) -- (J);
\end{tikzpicture}
```
Kruskal's Algorithm

Notes: "△ △" means step 1 to step 9