Problem 1:

we call n be the size of initial problems

so each time we split, we get:

$(1-\sigma).n$ and $\sigma.n$

Minimum depth appears when we go follow σn branch because size decrease very fast

$$n, \sigma n, \sigma^2 n, \dots, \sigma^k . n$$

leaf $\Rightarrow \sigma^k . n \le 1 \quad (1)$

$$\Rightarrow \sigma^k \le \frac{1}{n}$$

$$\Rightarrow \log(\sigma^k) \le \log\left(\frac{1}{n}\right)$$

$$\Rightarrow k . \log(\sigma) \le -1 . \log(n)$$

$$\Rightarrow k \geq \frac{-1 \log(n)}{\log(\sigma)}$$

because $\log(\sigma) < 0$ when $0 < \sigma < \frac{1}{2}$

so $\boxed{\text{mi depth is } \left\lceil \frac{-1 . \log(n)}{\log(\sigma)} \right\rceil}$

maximum depth:

we apply to (1) then

$$(1-\sigma)^k . n \le 1$$

$$\Rightarrow (1-\sigma)^k \le \frac{1}{n}$$

$$\Rightarrow k . \log(1-\sigma) \le \log\left(\frac{1}{n}\right)$$

$$\Rightarrow k \geq \frac{-1 . \log(n)}{\log(1-\sigma)}$$

KLONG

So max depth is $\left\lceil \frac{-1 \cdot \log(n)}{\log(1-\sigma)} \right\rceil$

Problem 2:

Pseudo code:

```
MIN-HEAPIFY(A, i)
   l = 2*i
   r = 2*i + 1
   min = i

   if l <= heap_size(A) and A[l] < A[min]
      min = l
   if r <= heap_size(A) and A[r] < A[min]
      min = r
   if min != i
      swap A[i] and A[min]
      MIN-HEAPIFY(A, min)
```

In worst case, we might have to fix up to the height of the tree, we know h ≈ O(logn)

So T(n) = O(logn) * some operations take constant time

In general, worst case is O(logn).

Problem 3:

COW,DOG,SEA,RUG,ROW,MOW.BOX,TAB,BAR,EAR,TAR,DIG,BIG,TEA, NOW,FOX.

Rearrange:

COW

DOG

SEA

RUG

ROW

MOW

BOX

TAB

BAR

EAR

TAR

DIG

BIG

TEA

NOW

FOX

We start from the last letter:

A: SEA, TEA

B: TAB

G: DOG, RUG, DIG, BIG

R: BAR, EAR, TAR

W: COW, ROW, MOW, NOW

X: BOX, FOX

After this we have:

SEA, TEA, TAB, DOG, RUG, DIG, BIG, BAR, EAR, TAR, COW, ROW, MOW, NOW, BOX, FOX

We start the middle letter:

A: TAB, BAR, EAR, TAR

E: SEA, TEA

I: DIG, BIG

O: DOG, COW, ROW, MOW, NOW, BOX, FOX

U: RUG

After this we have:

TAB, BAR, EAR, TAR, SEA, TEA, DIG, BIG, DOG, COW, ROW, MOW, NOW, BOX, FOX, RUG

The 1^{st} letter:

B: BAR, BIG, BOX

C: COW

D: DIG, DOG

E: EAR

F: FOX

M: MOW

N: NOW

R: ROW, RUG

S: SEA

T: TAB, TAR, TEA

After this we have:

BAR, BIG, BOX, COW, DIG, DOG, EAR, FOX, MOW, NOW, ROW, RUG, SEA, TAB, TAR, TEA

Problem 4:

a) $T(n) = 2T(n/4) + 1$

$a = 2$, $b = 4$, $f(n) = 1$

$\log_4 2 = \frac{1}{2}$

we observe:

$$f(n) = O(1) = O(n^{\frac{1}{2} - \varepsilon})$$

with $\varepsilon = \frac{1}{2}$

$$\Rightarrow T(n) = \Theta(n^{\frac{1}{2}}) = \Theta(\sqrt{n})$$

b) $T(n) = 2T(n/4) + \sqrt{n}$

$$= 2T(n/4) + n^{\frac{1}{2}}$$

$$n^{\log_b a} = n^{\log_4 2} = n^{\frac{1}{2}}$$

we see

$$f(n) = \sqrt{n} = \Theta(n^{\frac{1}{2}})$$

$$\Rightarrow T(n) = \Theta(n^{\frac{1}{2}} \cdot \log n)$$

$$= \Theta(\sqrt{n} \cdot \log n)$$

c) $T(n) = 2T(n/4) + n$

$n^{\log_b a} = n^{\log_4 2} = n^{\frac{1}{2}}$

we see

$f(n) = n = \Omega(n) = \Omega(n^{\frac{1}{2}+\varepsilon})$

$\Rightarrow \varepsilon = \frac{1}{2}$ and :

$a \cdot f(n/b) = 2 \cdot \frac{n}{4} = \frac{n}{2} < c \cdot n$

we choose $c = 0{,}75$

$\Rightarrow$ Master theorem valid

$\Rightarrow T(n) = \Theta(n)$

d) $T(n) = 2T(\frac{n}{4}) + n^2$

$f(n) = n^2$

$n^{\log_4 2} = n^{\frac{1}{2}}$

we see: $f(n) = n^2 = \Omega(n^{\frac{1}{2}+\varepsilon})$

here $\varepsilon = \frac{3}{2}$ and

$a \cdot f(\frac{n}{b}) = 2 \cdot \frac{n^2}{4^2} = \frac{n^2}{8}$

to have $\frac{n^2}{8} < c \cdot n^2$

we choose $c = \frac{1}{2}$ $\Rightarrow$ master theorem valid

$\Rightarrow T(n) = \Theta(n^2)$