Q1 a) The following non-deterministic procedure recognizes 2-NO-Majority, and obviously runs in polynomial time.

begin 2noMag (S)
set a := 0
set b := 0
for each x in S do
    either set a := a + s
    or set b := b + s
end for each
if a = b
    output Y
else
    output N
end if

b) If \( S^* \) contains both \((\exists s - b)\) and \((\exists s + b)\), then \( \exists s^*_1 \geq \exists s \) and \( \exists s^*_2 < \exists s \). Since \( S \) must be non-empty, we have \( \exists s^*_1 > \exists s^*_2 \). If \( S^* \) contains neither number, a symmetric argument shows \( \exists s^*_1 < \exists s^*_2 \).

c) \( \exists s^*_1 = \exists s + (\exists s - b) \)
\( \exists s^*_2 = \exists (S \setminus S') + (\exists s + b) \)
\( = 2 \exists s - \exists s' + b \)
Hence, if \( \exists s^*_1 = \exists s^*_2 \), then \( \exists s' + (2 \exists s - b) = 2 \exists s - \exists s' + b \)
\( \therefore 2 \exists s' = 2b \)
\( \therefore \exists s' = b \)
d) \((Q1\ could)\)

Define \( S^\ast_1 = S'' \cup \Xi S - b \) \( S^\ast_2 = S' \setminus S'' \cup \Xi S + b \)

Thus, \( S^\ast_1 \) and \( S^\ast_2 \) partition \( S^\ast \). Since \( \Xi S'' = b \),
we have

\[
\Xi S^\ast_1 = \Xi S'' + \Xi S - b = \Xi S
\]
\[
\Xi S^\ast_2 = \Xi S - \Xi S'' + (\Xi S + b) = \Xi S
\]

Hence \( \Xi S^\ast_1 = \Xi S^\ast_2 \) as required.

e) \n
1. \( S^\ast \) is a positive instance of \( 2\text{-}\text{No-Majority} \),

then, by part b), \( S^\ast \) can be partitioned into

\( S^\ast_1 \) and \( S^\ast_2 \) with \( \Xi S^\ast_1 = \Xi S^\ast_2 \) and \( S^\ast_1 \) containing

\( \Xi S - b \), but not \( \Xi S + b \). Now define \( S' \) as in

part c) so that \( \Xi S' = b \), whence \((S,b)\) is a

positive instance of \( \text{SUBSETSUM} \).

1. \((S,b)\) is a positive instance of \( \text{SUBSETSUM} \).

then \( S^\ast \) is a positive instance of \( 2\text{-}\text{No-Majority} \)

by d).

It is obvious that this reduction can be effected

in logarithmic space, since we need only store

carries in computing sums and differences.

Hence, \((S,b)\) \(\rightarrow\) \( S^\ast \) is a reduction. Since \( \text{SUBSETSUM} \)

is known to be \text{NP-hard}, and \( \text{SUBSETSUM} \subseteq^h \text{NP} \text{-hard} \),

\( 2\text{-}\text{No-Majority} \) is \text{NP-hard}.
Q2. a) Let \( S = 3 \). Note that, by assumption \( S_i \leq S_{i+1} + \ldots + S_p \). So let \( i \) be the largest number \((1 \leq i \leq p)\) s.t.

\[
\sum_{i=1}^{i} S_i \leq \sum_{i=j}^{p} S_j
\]

Now let \( S_1 = 3 \)

\[
S_2 = \sum_{i=1}^{i} S_i
\]

\[
S_3 = \sum_{i=1}^{i} S_i
\]

By construction, \( \sum_{i=1}^{i} S_i = \sum_{i=2}^{i} S_i + \sum_{i=3}^{i} S_i \), since this is just the statement (1). By assumption, \( \sum_{i=2}^{i} S_i \leq \sum_{i=1}^{i} S_i + \sum_{i=3}^{i} S_i \), since this is just the statement \( S_{i+1} \leq (S_1 + \ldots + S_i) + (S_{i+2} + \ldots + S_p) \). We need only show \( \sum_{i=1}^{i} S_i \leq \sum_{i=2}^{i} S_i + \sum_{i=3}^{i} S_i \).

Suppose \( i = p-1 \). Then, by assumption

\[
\sum_{i=1}^{p-1} S_i = \sum_{i=p}^{p} S_i \leq \sum_{i=1}^{p-1} S_i + S_p \]

and there is nothing to show. On the other hand, suppose \( i < p-1 \). Then, by maximality \( R_i \) in (1), we have

\[
s_1 + \ldots + s_{i+1} > s_{i+2} + \ldots + s_p
\]

But this is just the statement \( \sum_{i=1}^{i} S_i < \sum_{i=2}^{i} S_i + \sum_{i=3}^{i} S_i \), so we certainly have \( \sum_{i=1}^{i} S_i \leq \sum_{i=2}^{i} S_i + \sum_{i=3}^{i} S_i \) as required.
Q 2 could b)

Suppose the multiset $S$ is written in blocks on the input tape, with the integers (in binary) separated by \#. Write $S = \{s_1, \ldots, s_p\}$.

| bits of $s_1$ | $\#$ | bits of $s_2$ | $\#$ | $\#$ | bits of $s_p$ |

The following subroutine computes the $j$th bit and carry of the sum of all the $s_i$ except for $s_i$.

\[ \text{bit+Carry}(j, i) \]

Let $c' = \begin{cases} 0 & \text{if } j = 1 \\ \text{bit+Carry}(j-1, i) & \text{where } \text{bit+Carry}(j-1, i) = (b'', c'') \end{cases}$.

For each $i' = 1$ to $m$

Let $b'$ be the $j$th bit of $s_i$.

If $i = i'$ then

Let $c' = c' + b'$

End if

End for

Let $b$ be the least significant bit of $c'$ (i.e., shift $c'$ right)

Let $c = \lfloor c' / 2 \rfloor$ (i.e. shift $c'$ right)

Return $(b, c)$.

This clearly takes only logarithmic space, since the number of bits required for $c$ is $\log p$

Note that the recursive call space used by the recursive call can be recovered; we need only store the index $i$ and the carry $c''$. 

Now, to test if $S$ is a positive instance of $m$-NO-majority, just look for a number in $S$ greater than the sum of all the others. If none is found, reject; otherwise, accept.

$m$-Nimaj $(S)$

Let $S = s_1, \ldots, s_p$. Let $b$ be one greater than the maximum number of bits in any $s_i$ plus $p$.

For $i = 1$ to $p$

For $j = b$ to $1$ (i.e., in decreasing order)

Let $b$ be the $j$th bit of $s_i$.

Let $(b', c') = \text{bit} + \text{carry} (i, j)$

If $b > b'$ answer $N$ ($s_i > \text{sum of others}$)

If $b' < b$ set $j = 0$ ($s_i < \text{sum of others}$, so break out of inner loop)

end for $j$

end for $i$

answer $Y$ (No $s_i$ is greater than sum of all others)