```
0x788
--------
assert(0 == msg.value)
$s2 = ad_mask & c[0x4]
$s3 = intcall0(0x1879)
$s5 = intcall0(0x1883)
$s9 = intcall4($s2, 0x1891)
if (! $s9){
  $s10 = $m
  if (msize() >= $m){
    $s10 = msize()
  }
  m[$s10] = 0x0
  $m = 0x20 + $s10
  $s3 = $s10
  goto 0x1948
} else {
  $s10 = $m
  if (msize() >= $m){
    $s10 = msize()
  }
  m[$s10] = $s9
  $m = $s10 + (0x20 + (0x20 * $s9))
  $s5 = $s10
  $s9 = intcall1(0x18ea)
  $s6 = $s9
  $s7 = 0x0
  $s8 = 0x1
  while (0x1) {
    if ($s8 > $s6)
        break
    m[0x0] = $s8
    m[0x20] = 0x7
    if (ad_mask & s[sha3(0x0, 0x40)] == ad_mask & $s2){
      assert($s7 < m[$s5])
      m[0x20 + ($s5 + (0x20 * $s7))] = $s8
      $t = $s7
      $s7 = $s8
      $s8 = 0x1 + $t
      $t = $s7
      $s7 = $s8
      $s8 = $t
    }
    $s8 = 0x1 + $s8
  }
  $s3 = $s5
}
m[$m] = 0x20
$s4 = 0x20 + $m
m[$s4] = m[$s3]
$s4 = 0x20 + $s4
$s5 = 0x20 + $s3
$s6 = 0x20 * m[$s3]
$s10 = 0x0
while (0x1) {
  if ($s10 >= $s6)
        break
  m[$s10 + $s4] = m[$s5 + $s10]
  $s10 = 0x20 + $s10
}
return($m, ($s6 + $s4) - $m)
```