ESCAPE:

* Nhận thấy rằng các con chó phải ở lại sẽ đứng ở đáy tháp và đứng theo thứ tự nào cũng được, ta coi như chúng chịu trách nhiệm làm giảm độ sâu của hố và không quan tâm tới chúng nữa, còn các con chó thoát ra sẽ theo thứ tự tăng dần của 𝑏[.].Thật vậy, con chó có 𝑏[.] lớn nhất chắc chắn phải thoát ra cuối cùng, bởi nếu nó không thoát được thì tất cả những con khác cũng không thoát được nếu đứng vào vị trí của nó.
* Ta xếp các con chó tăng dần theo 𝑏[.] từ đỉnh tháp xuống đáy tháp.

𝑏1 ≤ 𝑏2 ≤ ⋯ ≤ 𝑏n

* Để tiện trình bày, ta gọi 𝑑(𝑖) là khoảng cách từ chân con chó 𝑖 tới miệng hố, theo cách sắp xếp ban đầu này thì:

𝑑(𝑖) = ℎ − ∑ 𝑎𝑗 (𝑛 ≥ 𝑗 ≥ 𝑖+1)

𝑑(𝑖) có thể âm trong trường hợp con chó 𝑖 đứng cao hơn miệng hố. Điều kiện để con chó 𝑖 thoát được là các con chó phía trên nó thoát được, đồng thời 𝑑(𝑖) ≤ 𝑏[𝑖].

* Ban đầu ta có một tháp, xét lần lượt các con chó từ 1 tới 𝑛 (từ đỉnh tháp xuống), nếu gặp một con chó 𝑖 không thoát được có nghĩa là tập các con chó {1,2, … , 𝑖} không thể thoát hết trong phương án tối ưu, chắc chắn ta phải bỏ lại ít nhất một con trong tập này, (những) con chó bị bỏ lại sẽ được chuyển xuống đáy tháp. Nếu con chó 𝑗 nào đó bị chuyển xuống đáy tháp (1 ≤ 𝑗 ≤ 𝑖) thì:
* Những con phía trên nó (1 … 𝑗 − 1) có 𝑑(. ) không đổi, khả năng thoát không bị ảnh hưởng Những con chó phía dưới nó (𝑗 + 1 … 𝑛) có 𝑑(. ) giảm đi 𝑎[𝑗], cơ hội thoát được tăng lên.
* Ta xét phép chọn tham lam: Chọn con chó 𝑗 (1 ≤ 𝑗 ≤ 𝑖) có 𝑎[𝑗] lớn nhất để chuyển xuống đáy tháp. Sau phép chuyển như vậy 𝑑(1 … 𝑗 − 1) không đổi, 𝑑(𝑗 + 1 … 𝑖 − 1) tăng lên, tức là những con chó 1,2, … ,𝑗 − 1,𝑗 + 1, … 𝑖 − 1 vẫn thoát được. Ngoài ra nếu 𝑗 ≠ 𝑖 thì con chó 𝑖 chắc chắn sẽ thoát được.
* Để duy trì các 𝑑(𝑖), trước hết ta tính 𝑄 = ∑ 𝑎𝑖 (1 ≤ i ≤ n) là tổng các độ cao đến lưng của tất cả các con chó. Khi xét tới con chó 𝑖, trước hết ta đặt 𝑄−= 𝑎𝑖 là tổng độ cao đến lưng của những con chó đứng dưới con chó 𝑖, khi đó ta có 𝑑(𝑖) = ℎ − 𝑄. Khi chuyển một con chó 𝑗 xuống đáy, ta chỉ việc tăng 𝑄 lên 𝑎𝑗 (một cách khác là giảm ℎ đi 𝑎𝑗 ) Như vậy ta không cần duy trì đại lượng 𝑑(𝑖) nữa mà tại mỗi bước lặp ta sẽ viết ℎ − 𝑄 thay vì 𝑑(𝑖). Để chọn nhanh 𝑎[.] lớn nhất, mỗi khi xét một con chó 𝑖, ta đẩy 𝑎𝑖 vào một hàng đợi ưu tiên, nếu con chó 𝑖 không thoát được, ta lấy phần tử lớn nhất trong hàng đợi ưu tiên ra, giá trị lấy ra sẽ được cộng thêm vào 𝑄 (hoặc trừ đi từ ℎ).