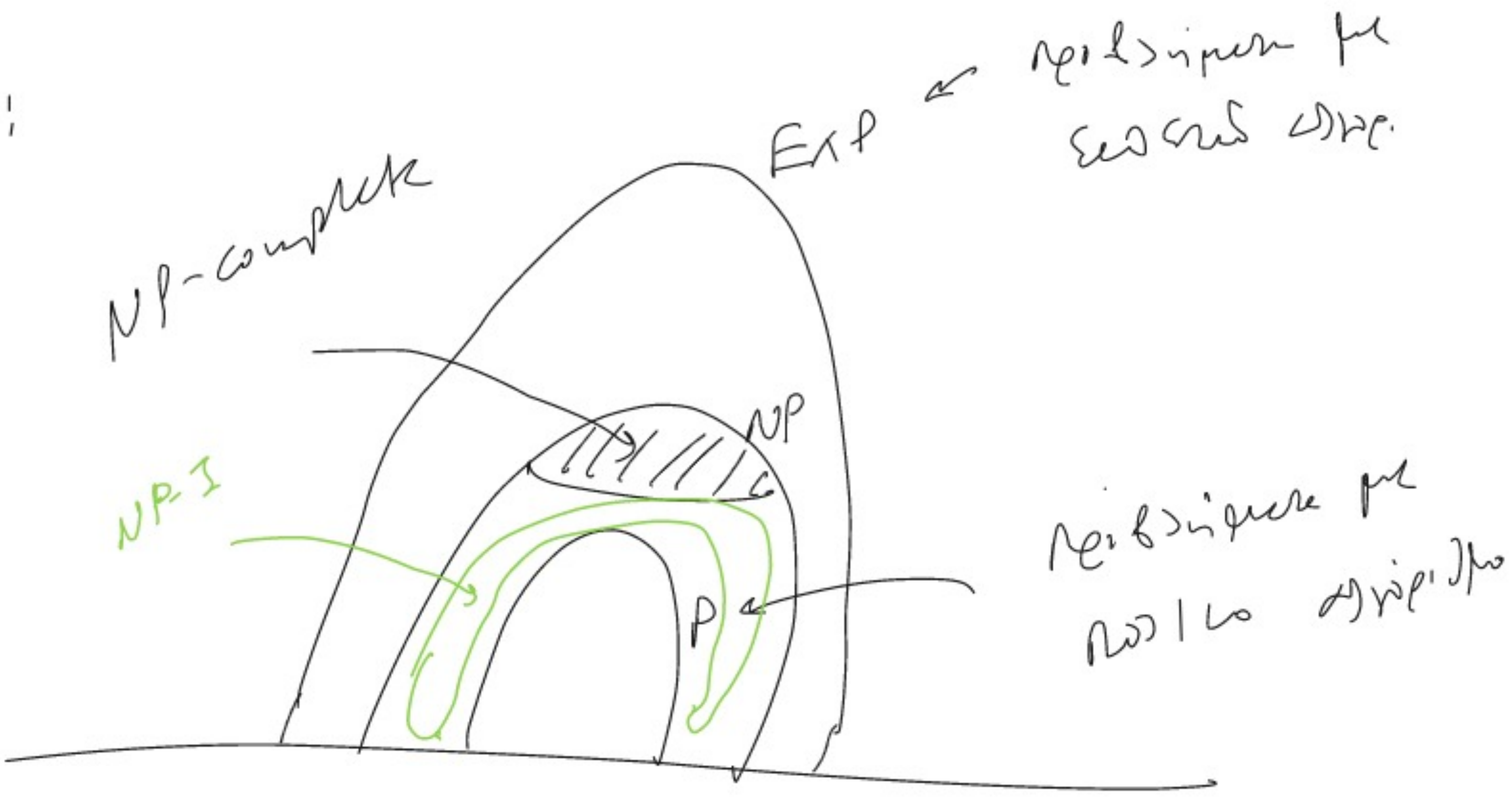
καλό μάντεμα!


Exp ← προβλήματα με εκθετικό χρόνο.
NP-complete
NP
NP-I
P ← προβλήματα με πολ/κο αλγόριθμο

κακό μάντεμα;

SAT : $(x \vee y \vee z) \wedge (x \vee \bar{y}) \wedge (\bar{y} \vee x \vee z \vee y)$

$x = 1$ | $z = 1$
$y = 1$

P = Μερικά προβλήματα για τα οποία υπάρχει αλγόριθμος για τη λύση σε λογικό χρόνο

NP = Μερικά προβλήματα για τα οποία υπάρχει αλγόριθμος για τη επαλήθευση σε λογικό χρόνο.

Millennium Prize Problems → 1000 000 $

- Υπόθεση του Riemann
- Εικασία του Poincaré ←

⋮

- $P \overset{?!}{=} NP$

ΝΑΙ! Είναι το ίδιο "δύσκολο" να λύσεις ένα πρόβλημα με το να επαληθεύσεις μια πιθανή λύση

ΟΧΙ! Υπάρχουν αποδεδειγμένα "δύσκολα" προβλήματα.

NP-complete προβλήματα: Τα πιο δύσκολα προβλήματα στην κλάση NP.

- Αναγωγή [illegible] $A \leq_p B$ αν υπάρχει

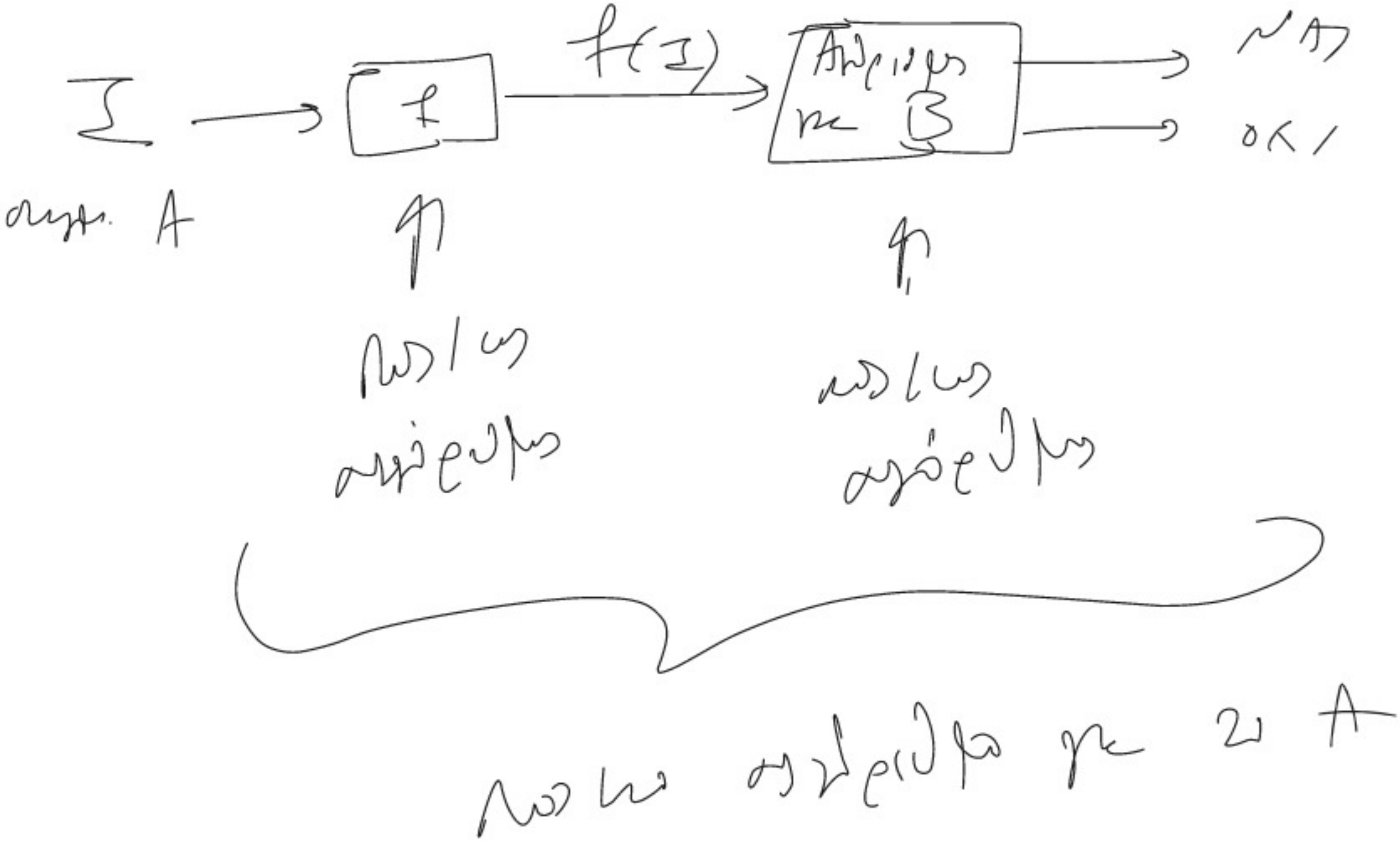
συνάρτηση f τ.ώ.:

$$I \longrightarrow \boxed{f} \longrightarrow f(I)$$

στιγμιότυπο για το A

στιγμιότυπο για το B

1. f πολ/κη
2. I ναι-στιγμιότυπο $\Longleftrightarrow$ f(I) ναι-στ.


I
στιγμ. A
f
f(I)
Αλγόριθμος
για B
ΝΑΙ
ΟΧΙ
πολ/κος
αλγόριθμος
πολ/κος
αλγόριθμος
πολ/κος αλγόριθμος για το A

B NP-complete αν:

$\forall A \in NP : A \leq_p B$

Θεώρημα Cook, Levin (1971):

κάθε πρόβλημα στην κλάση NP $\leq_p$ SAT

SAT $\leq_p B$

παραδείγματα αναγωγών:

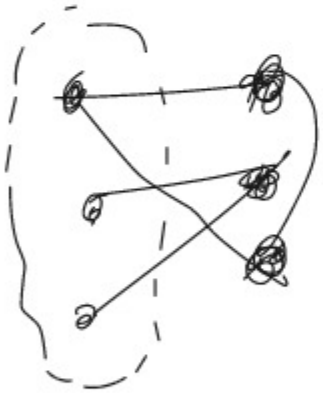
Κάλυψη Κορυφών $\leq_p$ Ακέραιο Γραμμ. Προγρ.

(G, k)

Ανεξ. Σύνολο $\leq_p$ Κάλυψη Κορυφών.

(G, k) $(G, n-k)$

$|S| \geq k$



S

$|V \setminus S| \leq k$

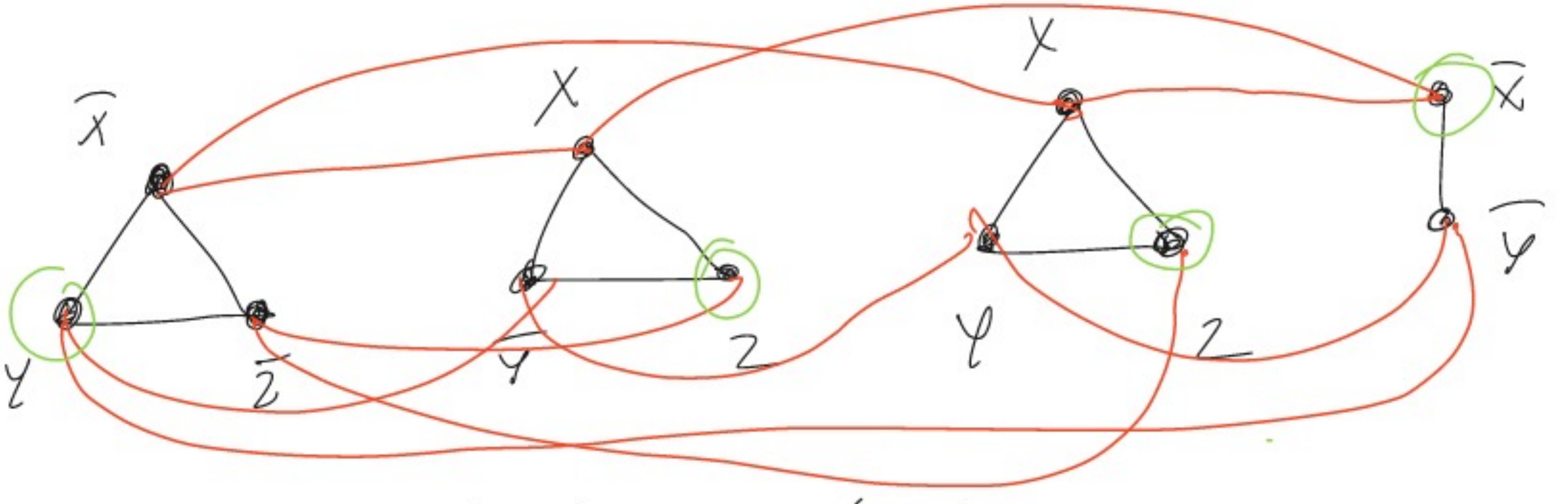
$$\left(SAT \leq_p 3SAT \right)$$

$3SAT \leq_p$ Ανεξ. Σύνολο.

είσοδ. 3SAT

$$\left[(\overset{1}{\bar{x}} \vee \overset{1}{y} \vee \bar{z}) \wedge (x \vee \bar{y} \vee \overset{1}{z}) \wedge (x \vee y \vee \overset{1}{z}) \wedge (\overset{1}{\bar{x}} \vee \bar{y}) \right]$$

είσοδ. Α.Σ.



$k = \#$ όρων (4)

- Είναι όλα τα προβλήματα NP-complete;

Richard Ladner (1975): $P \neq NP$ τότε υπάρχουν προβλήματα στην NP που δεν είναι NP-complete αλλά ούτε στην τάξη P.

- Ισομορφισμός Γραφημάτων

- Παραγοντοποίηση

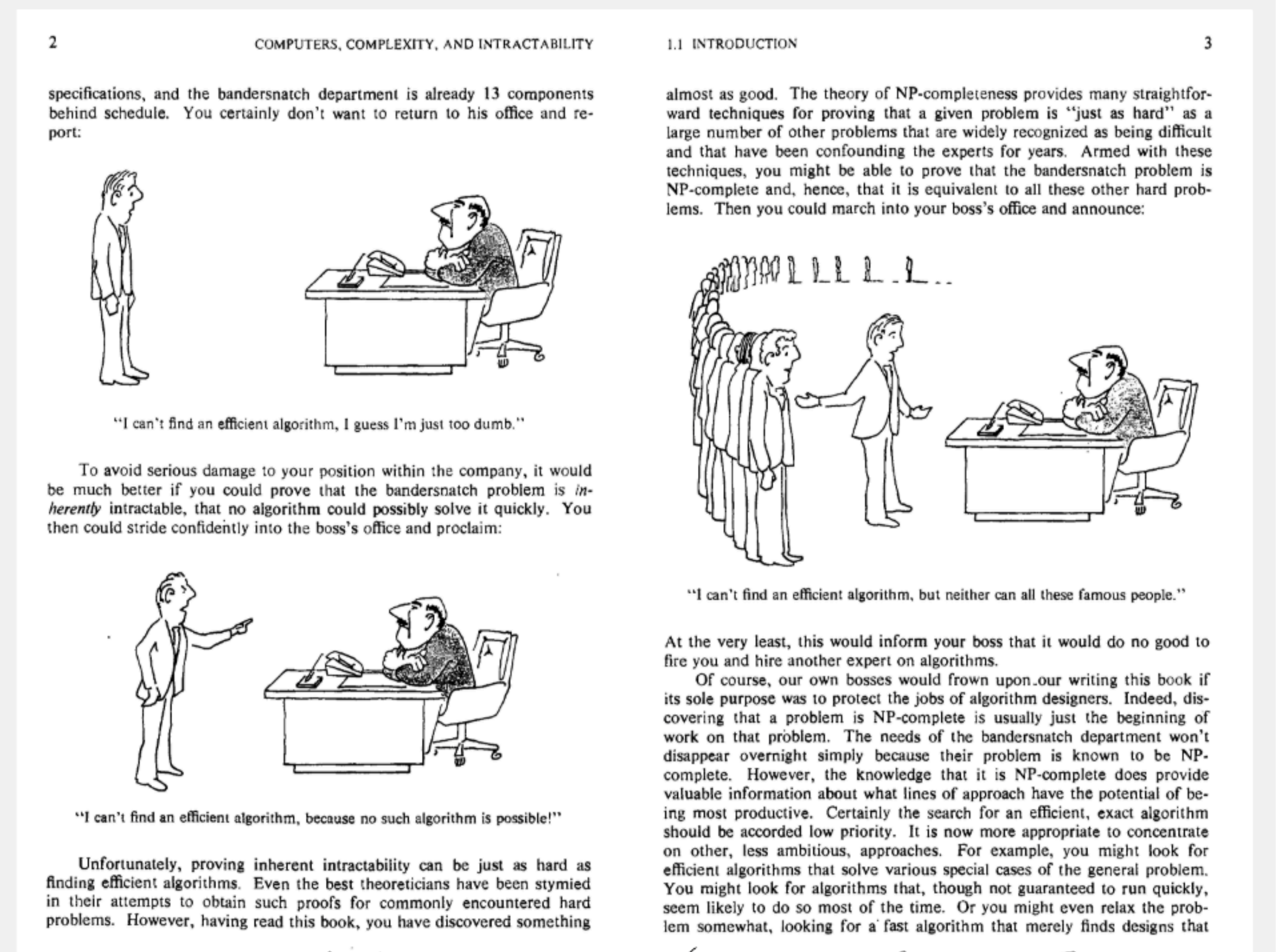
- Πόσο δύσκολο είναι να απαντηθεί το $P \overset{?!}{=} NP$;

Προσομοίωση

Διαγωνοποίηση

} μέθοδοι "μαύρο κουτί"

1975: Baker, Gill, Solovay: με χρήση "μαύρου κουτιού" απλά μεθόδους δεν μπορεί να απαντηθεί αν $P \overset{?!}{=} NP$.



specifications, and the bandersnatch department is already 13 components behind schedule. You certainly don't want to return to his office and report:

"I can't find an efficient algorithm, I guess I'm just too dumb."

To avoid serious damage to your position within the company, it would be much better if you could prove that the bandersnatch problem is *inherently* intractable, that no algorithm could possibly solve it quickly. You then could stride confidently into the boss's office and proclaim:

"I can't find an efficient algorithm, because no such algorithm is possible!"

Unfortunately, proving inherent intractability can be just as hard as finding efficient algorithms. Even the best theoreticians have been stymied in their attempts to obtain such proofs for commonly encountered hard problems. However, having read this book, you have discovered something

almost as good. The theory of NP-completeness provides many straightforward techniques for proving that a given problem is "just as hard" as a large number of other problems that are widely recognized as being difficult and that have been confounding the experts for years. Armed with these techniques, you might be able to prove that the bandersnatch problem is NP-complete and, hence, that it is equivalent to all these other hard problems. Then you could march into your boss's office and announce:

"I can't find an efficient algorithm, but neither can all these famous people."

At the very least, this would inform your boss that it would do no good to fire you and hire another expert on algorithms.

Of course, our own bosses would frown upon our writing this book if its sole purpose was to protect the jobs of algorithm designers. Indeed, discovering that a problem is NP-complete is usually just the beginning of work on that problem. The needs of the bandersnatch department won't disappear overnight simply because their problem is known to be NP-complete. However, the knowledge that it is NP-complete does provide valuable information about what lines of approach have the potential of being most productive. Certainly the search for an efficient, exact algorithm should be accorded low priority. It is now more appropriate to concentrate on other, less ambitious, approaches. For example, you might look for efficient algorithms that solve various special cases of the general problem. You might look for algorithms that, though not guaranteed to run quickly, seem likely to do so most of the time. Or you might even relax the problem somewhat, looking for a fast algorithm that merely finds designs that

Computers and Intractability (1979) Garey, Johnson