TEST [2]
[1]

1 SAT Chapter

2 Inapprox

The following is a randomized 0.5-approximation algorithm: for each vertex v flip a coin a fair coin to decide which half of the partition the vertex v goes into. By the method of conditional probabilities the algorithm can be derandomized. Hence there is a deterministic 0.5-approximation algorithm. DONE

3 Inapprox

The 15-puzzle [13] DONE

4 Intro

In addition, Saks & Santhanam [10] showed that if MCSP is **NP**-hard under other reductions (in between Karp and Turing reductions) then some other longstanding open problems would be solved. These results do not indicate .... DONE

5 In Preface- Section on $P, NP, \ldots$

In 1910 Pocklington [9] analyzed two algorithms for solving quadratic congruences mod m (note that m takes $\lg m$ bits to represent) and noticed that

- one took time proportional to a power of the $\lg m$, where as
- the other took time proportional to m itself or to $\sqrt{m}$.

In modern terms he was saying that one algorithm ran in polynomial (actually linear) time and the other algorithm took time non-polynomial (actually exponential). Unfortunately neither Pocklington, nor anyone else, pursued this distinction. Indeed, the notion that someone had earlier seen the distinction was not that well known until way after **P** and **NP** were defined (we did not know of Pocklington until we began working on this book). Pocklington paper is earliest reference to polynomial time that we know of.

In 1956 Kurt Godel postal mailed (there was no email back then) a letter to John von Neumann that, in modern terms, asked if a problem that is **NP**-complete is actually in **P**. Unfortunately John von Neumann never responded, and neither Godel nor anyone else pursued the distinction. Indeed, this letter itself only came to light in 1989. Urquhart [12] tells the entire story, plus why theoretical computer science did not emerge as a separate discipline until the 1960's.

In the early 1960's Cobham [4] and Edmonds [5] defined **P** and suggested it as a notion of efficient. Fortunately their definition was accepted and the stage was set for **NP** and **NP**-completeness.

DONE

6 Intro

Replace Google with

See Impagliazzo's survey [7] for more recent results. DONE

7 Exists R

Rename PSPACE-Hardness via QSAT to just PSPACE-Hardness

Put Exists R stuff into it at the end.

ADD to beginning of the section:

There are some problems that (1) are in **PSPACE**, (2) are **NP**-hard, (3) do not seem to be in **NP**, and (4) do not seem to be **PSPACE**-hard. In this section we define a complexity class to capture some of them.

REPLACE

The problem ETR is thought to be hard.

WITH

The problem ETR is thought to not be in **NP**.

ADD TO The THEOREM with Tarski and Canny a third

ETR is **NP**-hard. This is easy and left to the reader.

Showing that a problem is ER-complete does not just show that its unlikely to be in **P**, but that its also unlikely to be in **NP**. See Schaefer's [11] for a fuller discussion of the importance of ER-completeness.

Replace McDiarmid and Skerman with

Kang & Muller [8]

Problem 7.1. *Rectilinear Crossing Number (*RCN*)*

INSTANCE: A graph G and a number c.

QUESTION: Can G be drawn in the plane such that (1) every edge is a straight line, (2) every vertex has degree ≤ 2, and (3) there are at most c crossings? Schaefer [11], drawing heavily on Bienstock [3], showed that RCN *is* ER*-complete.*

DONE

8 NP-hardness via a misc

Bipartite and General Crossing Number

Exercise Show that....

needs a twiddle.

DONE

References

[1] S. Aaronson. A linear-optical proof that the permanent is #p-hard. *Electron. Colloquium Comput. Complex.*, page 43, 2011.
https://eccc.weizmann.ac.il/report/2011/043.

[2] A. Ben-Dor and S. Halevi. Zero-one permanent is $\#P$-complete, A simpler proof. In *Second Israel Symposium on Theory of Computing Systems, ISTCS 1993, Natanya, Israel, June 7-9, 1993, Proceedings*, pages 108–117. IEEE Computer Society, 1993.
https://doi.org/10.1109/ISTCS.1993.253457.

[3] D. Bienstock. Some provably hard crossing number problems. *Discret. Comput. Geom.*, 6:443–459, 1991.
https://doi.org/10.1007/BF02574701.

[4] A. Cobham. The intrinsic computational difficulty of functions. In *Logic, Methodology, and Philosophy of Science*, pages 186–192. North Holland, 1964.

[5] J. Edmonds. Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965.

[6] R. Greenlaw, J. Hoover, and W. Ruzzo. *Limits to Parallel Computation: P-Completeness Theory*. Oxford University Press, New York, 1995.
https://homes.cs.washington.edu/~ruzzo/papers/limits.pdf.

[7] R. Impagliazzo. Hardness as randomness: a survey of universal derandomization, 2003.
http://arxiv.org/abs/cs/0304040.

[8] R. J. Kang and T. Müller. Sphere and dot product representations of graphs. *Discret. Comput. Geom.*, 47(3):548–568, 2012.
https://doi.org/10.1007/s00454-012-9394-8.

[9] H. Pocklington. The determination of the exponent to which a number belongs, the practical solution of certain congruences, and the law of quadratic reciprocity. *Proceedings of the Cambridge Philosophy Society*, 16:1–16, 1910.

[10] M. Saks and R. Santhanam. Circuit lower bounds from NP-hardness of MCSP under Turing reductions. In S. Saraf, editor, *35th Computational Complexity Conference, CCC 2020, July 28-31, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 169 of *LIPIcs*, pages 26:1–26:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.
https://doi.org/10.4230/LIPIcs.CCC.2020.26.

[11] M. Schaefer. Complexity of some geometric and topological problems. In D. Eppstein and E. R. Gansner, editors, *Graph Drawing, 17th International Symposium, GD 2009, Chicago, IL, USA, September 22-25, 2009. Revised Papers*, volume 5849 of *Lecture Notes in Computer Science*, pages 334–344. Springer, 2009.

[12] A. Urquhart. Von Neumann, Godel and complexity theory. *Bull. Symb. Log.*, 16(4):516–530, 2010.
https://doi.org/10.2178/bsl/1294171130.

[13] Wikipedia. 15 puzzle.
https://en.wikipedia.org/wiki/15_puzzle.

[14] Wikipedia. Boolean satisfiability problem.
https://en.wikipedia.org/wiki/Boolean_satisfiability_problem.