

**Boston University**  
**Computer Science Convocation Address**  
*May 16, 2004*



**Harry R. Lewis**

Harvard College Professor; Gordon McKay Professor of Computer Science, Harvard University  
A.B., 1968, A.M., 1973, Ph.D., 1974, Applied Mathematics, Harvard University

---

It's a pleasure to see you here today. Congratulations to everyone being honored here! I am well aware that you see in me the last human being standing between you and your diplomas. I shall try to respect your priorities!

College commencements are one of the happiest imaginable occasions, the culmination of years of hard work, an opportunity for family reunions and heartfelt expressions of pride and joy. In America, college graduation also seems to be something of a ritual of entry into adulthood, the latest entry-to-adulthood ritual in any society I can think of.

One of the things that make college graduation such a happy occasion is, I suspect, that most such entry to adulthood rituals, be they religious events or junior proms, happen at an age when children are young enough so that they are a bit frightened about growing up and their loving parents don't think their children are ready to be considered grown-ups. Even on this occasion some of the fear on the part of the sons and daughters and the misgivings on the part of the parents still exists, though the graduates are mostly in their early 20s and the parents are, well, my age, heaven knows how old. With one daughter of my own in college and another about to start graduate school I would be the last to suggest that parents and their children have had enough of each other by now. But at this stage of life both parties to the deal are ready to be glad, really overjoyed deep down, that anything resembling childhood can finally be declared to be over.

Among rituals of passage, college graduation can also be one of the least memorable. Take lots of photographs, because my guess is you are so excited that the neural imprints of what is going on around you are not being laid down very effectively. Such limitations of memory can also be caused by tensions about whether your mother will like your boyfriend and whether your father will get off your case about the job you took, or perhaps haven't been offered yet – but will be any day now! So parents too need to relax and enjoy it. The moment won't come again and whatever you parents are worried about today can be can certainly be worried about equally well out later on. Better no memories of today than bad ones.

Graduation speeches are a terrible genre. I remember absolutely nothing about the speaker at my own college graduation, except who it was, and that he too was speaking in the rain. I don't even remember that about my high school graduation. But my college graduation speaker was a head of state, probably signed up two years in advance. By the time the date came around, his reputation as an enlightened leader was badly frayed, and he was deposed shortly thereafter. That left some doubt in our minds as to whether we should really pay attention to the advice he offered us about the future. I am sure that the people who picked him have, like me, mercifully forgotten the speech and everything else that happened that day.

There will be no reason for you to remember me either, but I do hope you will remember some people who were important to you and to your education.

Before all others, remember those who got you here. Of course you got yourself here; no one graduates from BU without earning it. You've had the benefit of some outstanding faculty and I know they teach a tough curriculum. So you certainly have yourselves to thank.

But your parents got you here too. Oh I know they caused you all kinds of grief, and if you are the way I was when I graduated from college you may be conscious today of the ways in which your parents could be improved if you could slightly remodel them. But remember that you are their pride and joy and that they made sacrifices to make this day possible for you. And there are others too, grandparents, and brothers and sisters perhaps to whom you looked up, or who pushed you by looking up to you.

I need to pause here to remember and thank my own father. He was born and grew up in East Boston, the son of immigrants recently arrived from across the Atlantic, just as I imagine more than one or two of you graduates came here from oceans away, or your parents did. Boston University was the place that gave him a shot at a life better than the one his parents had had. I'll bet that story is being repeated here today for many of you. So parents, a nod to you. I know you deserve it, and part of what you are feeling in your sons' and daughters' triumphant day today is how far you yourselves have come. Congratulations to you too. And a nod to Boston University too, for the thousands of door-openings it has made possible for more than a century.

Now a word or two about the field. Computer science is a wonderful subject because it is so new. We still don't know very much about it. It's not like civil engineering, where the principles have been known for a long time and if they build a building and the roof leaks, it's not because the theory of roofs isn't fully understood yet. Years ago I visited Beauvais Cathedral in France, a spectacular construction in the high Gothic style. But it's only partly there – a big part of it fell

down in 1284 because no one had ever built quite as tall a nave and the engineering aspects of a structure that high had not been worked out yet. Buildings don't tend to fall down any more. Yet computer systems almost always fail when they are first built, and often keep failing in new and annoying ways years after they have been released. (By the way, I love that phrase, "released," which is what many of you will be doing to the software you create. Software systems are apparently like caged animals that are sometimes let loose. What does that make software engineers?) In any case, in computing there are lots of unsolved problems and there are still huge opportunities for making a difference. The field is still in its infancy.

But exciting as computing is, it is also a deeply frustrating business. Almost everything you learn about the field in college seems to fall into one of two categories. Half of it is stuff that seems so incredibly clever, complicated, and incomprehensible that you can't imagine how anyone could have thought of it or built it. This half of the subject makes you feel small, intimidated by the brilliance of the other people in the field. The Turings and the Knuths and the Tarjans and the Rivests, those are really smart people. Most of them are still alive, too! The people who design and implement hugely complex systems, processor chips and operating systems and Google's file system, and get those things done more or less on schedule, my goodness that is impressive. I wouldn't know where to begin to make something that big, yet we take it for granted every day that such things work, at least most of the time.

The other half of the subject is full of things that seem so obvious that it makes you mad they couldn't have stayed undiscovered long enough for **you** to be the one to make your fame and fortune by inventing them. Any fool could have thought of packet switching or Huffman coding. Did someone really get credit for inventing relational databases? My favorite one of these right now is the algorithm for figuring out what text message you are typing on a telephone keypad even though you only hit the key for each letter once. It doesn't look like that is possible but it is, you just have to type ahead a little bit before the phone figures out what the key you typed two or three keys ago was supposed to have designated. Not hard to do once you think about it, but it is neither you nor I who is going to be remembered for thinking of it.

Computers keep getting smaller and cheaper, and I'd suggest you try to bear that in mind. If I were your age today, I'd be thinking about things you can do with really small, really cheap computers with sensors and short-range radios. These are not going to have megabytes of memory and gigabytes of disk in them. They are going to run on watch batteries, so they are going to be slow and not very bright. It will be possible to make these things for a buck or two before long. You will be able to buy them the way you now buy flashlight batteries or paper plates. They will

be sold in drug stores and hardware stores. Without wires or programming, they will be set up to do useful things. I just don't know what. Ten years from now, it will be obvious what we should have thought of today as useful applications of these little things.

There are a million things you could do with such devices, many of them silly but some of them valuable. (And some, no doubt, both silly and valuable.) I heard a good idea this week. It's not too hard to build one of these with a sensor that can tell if someone is nearby (or at least a dog or some other warmblooded animal). We all immediately think of the burglar alarm application, but they had better work really well if they are going to call the police. There are plenty of other applications where the stakes are lower. The payoff may be less too, but if the things are cheap that may not matter. For example, if you are building a huge skyscraper, you want the elevators to be as efficient as possible. So why do people have to push a button to call the elevator? In many big buildings the only reason for going near the elevators is to take one, so it should be possible to save a few seconds by sensing that someone is approaching the elevator and summoning the elevator as soon as that happens.

The nice thing about this idea is that it's not the kind of problem we usually try to solve. Usually we are worried about writing perfect code because we set a high store on computer systems working correctly all the time, and I have already complained that they usually don't. But these little things that summon elevators or tell you the humidity in the middle of a wheat field or monitor the air quality at your landfill don't have to work perfectly. If there are enough of them, they can reach a consensus, and even if they malfunction occasionally the worst that happens is that someone has to push the elevator button or the irrigation sprinklers go on when they shouldn't or something like that. It doesn't really matter as long as these things work most of the time. Wherever you work there will be opportunities to think of your own good ideas or at least to help others' become a reality.

So the point is, looking back on everything that has happened since I was hearing a speech at my college graduation. Computers have gotten a whole lot cheaper but that trend isn't going to stop any time soon and it is worthwhile trying to think ahead to what really cheap computers could do for us.

One thing is for sure. Dot-com bust or no, the revolution is not over. The world is going to change because of information technology as much in the next decade as it did in the last. Don't get discouraged if you don't find your niche in it right away. You have gotten a good education and you are in the right field. Go forth and prosper. And send your professors a postcard or an email along the way. They would like to hear from you because your successes and trials are also theirs. Congratulations and best wishes.

## Short Biography of Speaker

Harry R. Lewis is a Gordon MacKay Professor of Computer Science at Harvard University. He is Boston raised, Harvard educated (AB 1968, AM 19, Ph.D. 1974) and lives in Brookline, MA.

Professor Lewis has worked extensively on the algorithmic solvability of logical, computational, and combinatorial systems, attempting to clarify the relations between them and to identify their common characteristics. He is well-known for his research in mathematical logic and theoretical computer science. His work centers on classifying unsolvable and hard to solve problems and in identifying common characteristics of such problems.

Professor Lewis is actively involved in the use of computers in education, and his books have had a significant influence on the teaching of the foundations of computer science to undergraduates. He has written several important textbooks and research monographs in this area. The most influential of these is his text "*Elements of the Theory of Computation*," with Christos Papadimitriou, which is a widely used text in theory of computation classes at Boston University and throughout the country.

For eight years, from 1995 until last July, Harry Lewis was the Dean of Harvard College. During his tenure as Dean he affected many changes and important institutional reforms in the College. Starting in Spring 2004, Professor Lewis has been a Visiting Professor in our Department and will be returning to Harvard University next year.